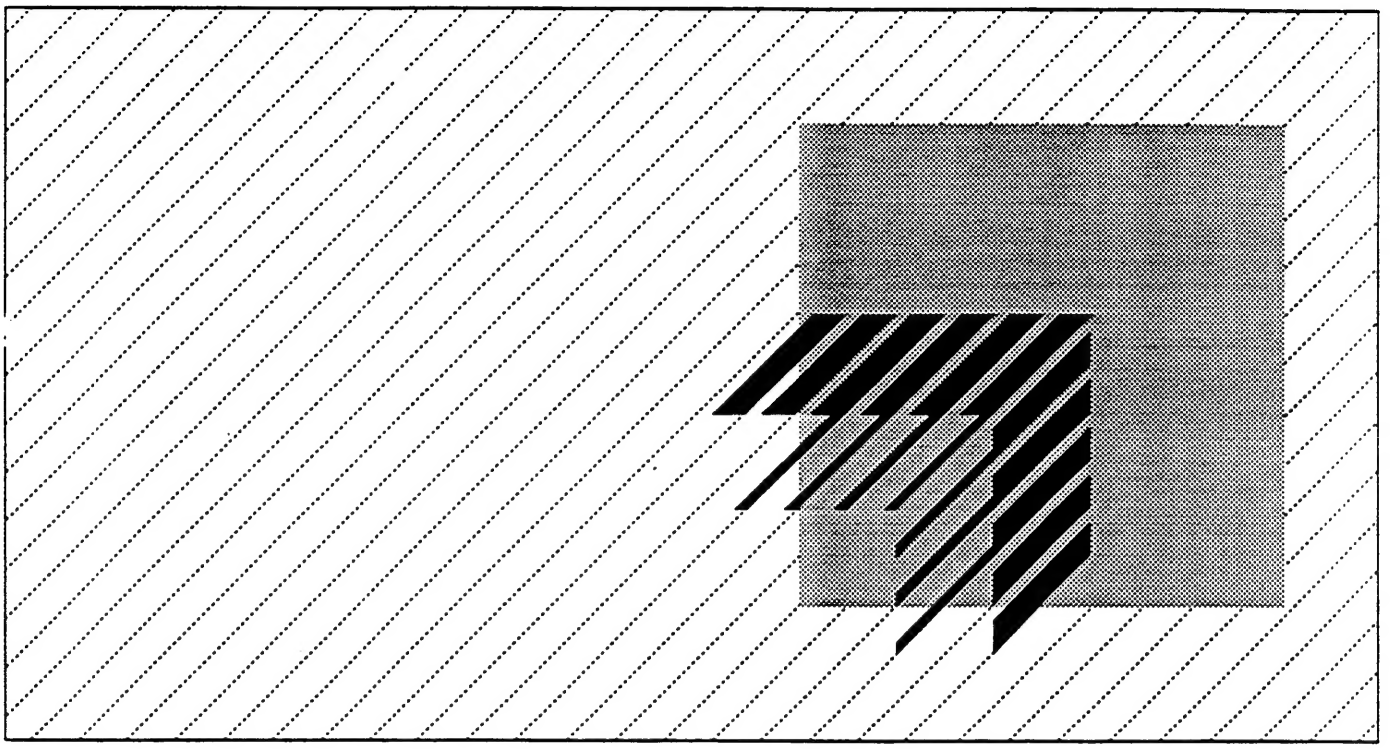
SIEMENS
NIXDORF



SINIX-SV.4

Systenverwaltung mit SINIX V5.4 und UNIX V.4

(Bestell-Nr.: U54184-J-Z103)

Mehr Wissen. Mehr Chancen. Mehr Erfolg.
Siemens Nixdorf Training Center.

SINIX-SV.4

Kursorganisation

H. HENKEL			Kursleiter
09:30			Kursbeginn
10:00	bis	10:30	Vormittagspause
12:00	bis	13:00	Mittagspause
	bis		Nachmittagspause
15 30			Kursende
≈ 13:00			ungefähres Kursende am letzten Kurstag

Allgemeines

0201 81320 -148 - 149	Rufnummer der Schule

Sonstige Bemerkungen zur Kursabwicklung:

Wir sind ständig bemüht, unsere Unterlagen zu verbessern und sie dem veränderten Bedarf anzupassen.
Um dieses Ziel zu erreichen, bitten wir Sie, uns Ihre Anregungen, Hinweise und Verbesserungsvorschläge zu der vorliegenden Lehrunterlage mitzuteilen.

Für Ihre Mitarbeit bedanken wir uns.

Vervielfältigung dieser Unterlage sowie Verwertung ihres Inhalts sind unzulässig, soweit nicht ausdrücklich zugestanden. Technische Änderungen vorbehalten.

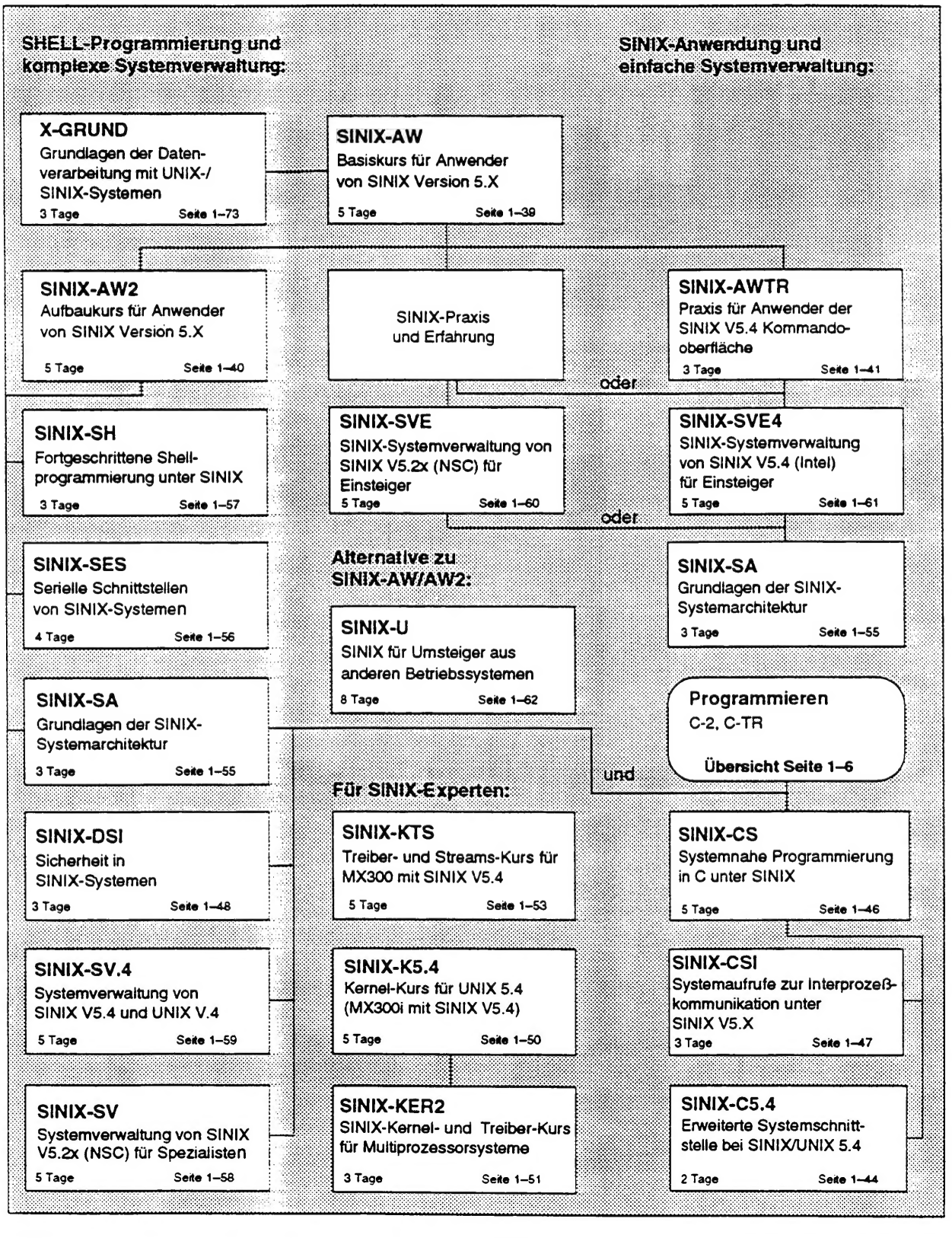
Siemens Nixdorf Informationssysteme AG

Herausgegeben von
Siemens Nixdorf Informationssysteme AG
Training Center
Postfach 830951,
Otto-Hahn-Ring 6
8000 München 83

Printed in the Federal Republic of Germany

SINIX-Betriebssystem

Sie wollen auf der Betriebssystem-Ebene arbeiten?
Wir bieten Ihnen dazu Kurse für Einsteiger, Fortgeschrittene und Spezialisten, die Konzepte für Systemsicherheit entwerfen, die SINIX-Systemschnittstelle nutzen und selbständig mit dem Kernel arbeiten.



Systemverwaltung von SINIX V5.4 und UNIX V.4

SINIX-SV.4

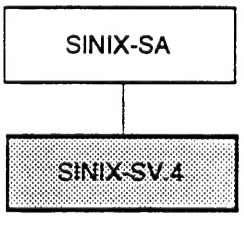
5 Tage

Zielgruppe

Systemverwalter
Systemspezialisten,
„Umsteiger" von V5.2x auf V5.4

Voraussetzung

Erfahrung in der Shell-Programmierung und
viel UNIX/SINIX-Erfahrung



Lernziel

Sie können SINIX V5.4-(Intel-) Rechner eigenverantwortlich verwalten.
Sie können
- SINIX V5.4 konfigurieren
- Platteneinteilung vornehmen
- Benutzer einrichten
- Den Hardwareausbau prüfen und optimieren
- Im Fehlerfall die richtigen Maßnahmen treffen
- Bildschirmarbeitsplätze und Drucker konfigurieren

Sie kennen ein Konzept zur Datensicherung und können
das Betriebssystem auf Magnetbandkassette sichern und wieder restaurieren.
Sie kennen den Ablauf der Betriebssysteminstallation und der Hardwaremonitor des MX500-90.

Kursinhalt

Dateibaum von SINIX/UNIX 5.4
Dateisystemtypen s5, ufs, bfs
Gerätedateien für Platten, Disketten, Streamer
Platteneinteilung „slices"
Zusätzliche Festplatte(n) konfigurieren
Dateisysteme erzeugen mit „mkfs"
Systemstart und rc-Verzeichnisse
Run-Level und Systemstopp
Terminalverwaltung
Benutzerverwaltung
Softwareinstallation
Bediensysteme „admin", „sysadm"
Zusätzliche Festplatte(n) konfigurieren
Datensicherung, Restaurierung
Installation des Betriebssystems
Notreparatur des Systems
Hardwaremonitor des MX500-90
Regelmäßige Aufgaben des Systemverwalters
Systemtuning
Praktikum auf SINIX V5.4

Hinweis
SINIX V5.4 ist das Betriebssystem der MX300- und MX500-Rechner mit Intel-Prozessor

---

Inhalt

	Seite
Hardwareaufbau und einfache Systemverwaltung	**L 1 - 1**
Rechnerleistung	L 1 - 1
Aktuelle SINIX-Rechner und Versionen	L 1 - 2
- MX500- Die wichtigsten Daten im Vergleich	L 1 - 4
Interner Aufbauf der SINIX-Rechner	L 1 - 5
- Parallelprozessorsystem MX500-90	L 1 - 6
- TACSI	L 1 - 7
- Externe SCSI-Geräte	L 1 - 8
Benutzer und Benutzergruppen	L 1 - 9
- Menügeführte Systemverwaltung	L 1 - 11
- Neuen Benutzer eintragen	L 1 - 13
- Wichtige Tasten von Alpha-COLLAGE	L 1 - 14
- Alpha-COLLAGE Sondertasten	L 1 - 15
- /etc/passwd, /etc/shadow	L 1 - 16
- /etc/group	L 1 - 17
Kommandos zur Benutzerverwaltung	L 1 - 18
- Das "passwd"-Kommando	L 1 - 19
- /etc/default/login, /etc/default/su	L 1 - 20
- Anmelden über X-Terminals	L 1 - 21
Weitere Kommandos zur Systemverwaltung	L 1 - 22
- Korn Shell: ksh	L 1 - 23
- Kommandos, Tagesmeldung, Eröffnungsbildschirm	L 1 - 24
- t-Bit bei "/tmp" und "/var/tmp"	L 1 - 25
Konfigurierung der Peripherie	L 1 - 28
- Systemdateien zur Terminalverwaltung	L 1 - 30
- gettydefs oder ttydefs	L 1 - 31
- terminfo statt termcap	L 1 - 32
- Eigenschaften von "terminfo"	L 1 - 33
Die SINIX Druckerverwaltung	L 1 - 34
- Die CONFIG-Datei	L 1 - 35
- Neustart des Spoolsystems mit neuer CONFIG	L 1 - 36
Dateibaum, Plattenbereiche, Dateisysteme	**L 2 - 1**
UNIX/SINIX-Stammbaum	L 2 - 1
X/OPEN-Standards	L 2 - 2
Dateibaum in SINIX 5.4/UNIX V.4	L 2 - 3
- Dateibaum in SINIX V 5.4	L 2 - 4
- Das "/var"-Dateisystem	L 2 - 5
- Das "/usr"-Dateisystem	L 2 - 6
- Shared Libraries in V5.4	L 2 - 7
- Einfache symbolische Verweise	L 2 - 8
Logische Platteneinteilung	L 2 - 9

Inhalt

	Seite
Das Verzeichnis /dev	L 2 - 11
- Gerätedateien für Plattenbereiche	L 2 - 12
Plattentypen und Plattendaten	L 2 - 13
Systemressourcen beim MX300	L 2 - 15
"block" und "raw" Gerätedateien	L 2 - 16
Was ist ein Dateisystem ?	L 2 - 18
- Dateisystemtypen	L 2 - 19
- Dateisystem "ufs"	L 2 - 20
Platteneinteilung: /etc/vfstab	L 2 - 22
Kommandos für Dateisysteme	L 2 - 23
- fsck, mount, umount	L 2 - 24
- mount, umount	L 2 - 25
- Verteilte Dateisysteme (NFS)	L 2 - 27
- Dateisysteme neu erzeugen	L 2 - 28
- mkfs-Kommando	L 2 - 29
- Anzahl der "inodes"	L 2 - 30
Zusätzliche Platte konfigurieren	L 2 - 34
- disksetup	L 2 - 35
Platteneinteilung nachträglich ändern	L 2 - 36
- Dateiformat: "edvtoc -f" und "prtvtoc -f"	L 2 - 37
"swap"-Bereiche konfigurieren	L 2 - 38
Sinnvolle Plattenaufteilung	L 2 - 39
Systemaktivitäten messen	L 2 - 40
- Beispiele für sar	L 2 - 42
- Auswertung der Messdateien	L 2 - 43
truss: Systemaufrufe protokollieren	L 2 - 44
Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben	**L 3 - 1**
Boot von Diskette oder Platte	L 3 - 1
- boot-Parameter	L 3 - 1
Meldungen beim Systemstart	L 3 - 3
- klog.msg für MX500	L 3 - 4
Feststellen von Hardwarefehlern	L 3 - 5
Bedienfeld MX300	L 3 - 6
Bedienfeld MX500	L 3 - 7
Arbeitsspeicherabzüge	L 3 - 9
"run-level", init	L 3 - 11
- run level 0,1,2 und 3	L 3 - 12
- /etc/inittab	L 3 - 13
- inittab verändern	L 3 - 14
autopush, /etc/ap/chan.ap	L 3 - 15
Betriebssystem herunterfahren	L 3 - 16
rc-Dateien in V5.4	L 3 - 17

Inhalt

	Seite
Fehler im Dateisystem	L 3 - 19
Einbenutzerbetrieb: run-level 1	L 3 - 20
"fsck" selbst aufrufen ?	L 3 - 21
- lost+found	L 3 - 22
Regelmäßige Kontrollaufgaben	L 3 - 23
Weitere Protokolldateien	L 3 - 24
Prozeduren automatisch starten	L 3 - 25
- Das "at"-Kommando	L 3 - 26
Systemzeit, Weltuhr, Hardwareuhr	L 3 - 28
- TZ-Variable	L 3 - 29
- Systemzeit und Hardwareuhr stellen	L 3 - 30
Datensicherung	**L 4 - 1**
Geräte zur Datensicherung	L 4 - 1
Arbeiten mit Magnetbandkassette	L 4 - 2
Pufferung	L 4 - 3
Grundprinzip von tar, cpio und ufsdump	L 4 - 4
"mt": Bandkontrolle	L 4 - 6
Bänder kopieren ?	L 4 - 8
Sicherungskommandos	L 4 - 9
cpio	L 4 - 10
- Dateibaum kopieren	L 4 - 11
Gerätedateien für Disketten in V5.4	L 4 - 12
Datenaustausch mit SINIX V5.2x	L 4 - 13
Das tar Kommando	L 4 - 14
- tar Aufruf	L 4 - 15
- Die Datei "/etc/default/tar"	L 4 - 16
- tar	L 4 - 17
dd	L 4 - 18
Sicherungskonzept	L 4 - 19
Arbeitsschritte zur Systemsicherung	L 4 - 20
- Grundsystem sichern	L 4 - 21
Aufgaben	L 4 - 22
Sicherung ganzer Dateisysteme	L 4 - 24
- ufsdump / ufsrestore	L 4 - 25
- Interaktives "ufsrestore"-Kommando	L 4 - 27
Spiegelplatten (VPSS)	L 4 - 28
Installation, Notreparatur, Systemparameter	**L 5 - 1**
Key-Diskette	L 5 - 1
Vorinstalliertes Betriebssystem	L 5 - 2
sysadm in V5.4	L 5 - 3
Installation von Softwareprodukten	L 5 - 4

Inhalt

	Seite
/etc/device.tab	L 5 - 5
Software-Pakete	L 5 - 6
Erzeugen von Softwarepaketen (für Entwickler)	L 5 - 8
- Datei: pkginfo	L 5 - 10
Installation auf dem MX300	L 5 - 11
Installation auf dem MX500-90	L 5 - 12
Notreparatur	L 5 - 14
Systemgrenzen, Systemparameter	L 5 - 17
- Einige Systemparameter	L 5 - 18
Besonderheiten des MX500	**L 6 - 1**
Bedienung des Hardware-Monitors	L 6 - 1
Gerätenamen und Nummern	L 6 - 2
bsu-Programme booten	L 6 - 3
Weitere Besonderheiten des MX500	L 6 - 4
Anhang	
Lösungen	**1 - 12**

Hardwareaufbau und einfache Systemverwaltung

Rechnerleistung

Die Leistung eines Mehrplatzrechners hängt ab von folgenden Leistungsmerkmalen :

- **Prozessorleistung** (wird oft in "MIPS" angegeben)
- **Arbeitsspeicher**
- **Festplatten** und **Plattensteuerungen** (Controller)
- Schnittstellen zur Peripherie (z.B. E/A-Prozessoren)
- Interne Datenübertragung (Bussysteme)

Prozessorleistung:

- **Taktfrequenz**

  Je höher die Taktfrequenz, desto höher ist theoretisch die Rechenleistung. Bei Taktfrequenzen ab 25 MHz muß der Prozessor jedoch häufig auf Befehle oder Daten warten, die erst aus dem Arbeitsspeicher geholt werden müssen.

  Moderne Prozessoren besitzen daher:

- **Prozessor-Cache**

  Schnelle Zwischenspeicher für Befehle und Daten zum Vermindern der Arbeitsspeicherzugriffe

- **Second Level Cache**

  Zusätzlicher schneller Zwischenspeicher auf der CPU-Baugruppe

- **Pipeline**

  Der Prozessor liest Befehle bereits im voraus und holt sich die zugehörigen Daten

Aktuelle SINIX-Rechner und Versionen

SINIX V5.2x

MX300-15/-30:

Prozessor: NSC 32532, Taktfrequenz 25 MHz (max. 16 MB Speicher)
Leistung: 8 MIPS

MX500-75: Multiprozessor

Prozessoren: NSC 32532, Taktfrequenz 25 MHz (max. 64 MB Speicher)
bis zu 8 Prozessoren
Platten: interne ESDI-Platten am Multibus I (1,6 MB/s)
Leistung: max. 64 MIPS (8 MIPS pro Prozessor)

SINIX V5.4

MX300-45/50:

Prozessor: INTEL 80486, Taktfrequenz 25 MHz
Leistung: 15 MIPS

MX300-60:

Prozessor: INTEL 80486, Taktfrequenz 33 MHz, 4 Mbit-Speicherchips
Leistung: 15 MIPS

MX300-60/3:

wie MX300-60 aber mit 128 KB Second Level Cache
Leistung: 17 MIPS

MX300-60/5:

Prozessor: INTEL 80486, Taktfrequenz 50 MHz
256 KB Second Level Cache
Leistung: 32 MIPS

MX500-90: Multiprozessor

Prozessoren: INTEL 80486, Taktfrequenz 25 MHz,
2 bis 7 Prozessoren, 4 Mbit-Speicherchips (max. 128 MB Speicher)
128 KB Second Level Cache je Prozessor

Platten: interne SCSI-Platten am speziellen synchronen SCSI-Bus (4 MB/s)

Bus-Systeme:

- Systembus: 52 MB/s
- synchrone SCSI-Kanäle für interne Festplatten: jeweils 4 MB/s
  (2 oder 4 Kanäle im Grundschrank, bis zu 4 für den Erweiterungsschrank)
- asynchroner SCSI-Bus intern: 1,5 MB/s
- bis zu 3 Multibus I für Ein/Ausgabe: jeweils 1,6 MB/s

Achtung: Diese Angaben gelten nur für "echte" MX500-90.
MX500-75, die auf Intel-Prozessor umgerüstet wurden, haben keinen synchronen SCSI-Bus und betreiben ihre Platten weiterhin am Multibus I.

In Vorbereitung:

Dualprozessor-Boards mit 50 MHz (intern) und 512 KB Cache für MX500-90/2.
Die restliche Hardware bleibt gleich.

MX500 - Die wichtigsten Daten im Vergleich

MX500 Modell 75

Prozessoren (NSC)
- Dualprozessor-Boards
- NSC 32532, 25MHz
  Cache je Prozessor:
  On-chip 1,5 Kbyte,
  On-board 64 Kbyte
- Bis zu 8 Prozessoren
- Prozessorleistung 16 - 64 Mips

Hauptspeicher (1 Mbit)
- 16 bis 64 Mbyte

5 1/4"-Festplatten
- Anschluß über ESDI-Controller an Multibus I
- 380/760 Mbyte Laufwerke
- Bis zu 6 ESDI-Controller

- Max. 12 Laufwerke
  (6 im GS, 6 im ES)
- Bis zu 9,1 Gbyte

5 1/4" Floppy Disk

SINIX V5.2x
- UNIX V.2 für Multiprozessorsystem

MX 500 Modell 90

Prozessoren (Intel)
- Monoprozessor-Boards / später Dualproz.
- Intel 80486, 25 MHz
  Cache je Prozessor:
  On-chip 8 Kbyte,
  On-board 128 Kbyte / später 512 KB
- Bis zu 7 Prozessoren
- Prozessorleistung 30 - 105 Mips

Hauptspeicher (4 Mbit)
- 32 / 64 / 128 Mbyte

5 1/4"-Festplatten mit Controller
- Anschluß über SCSI-Systembus-Adapter an schnellem Systembus
- 760 Mbyte/1,5 Gbyte Laufwerke
- 1 oder 2 SCSI-Systembus-Adapter jeweils mit 2 oder mit 4 Kanälen
- Bis zu 24 Laufwerke
  (12 im GS, 12 im ES)
- Bis zu 36 Gbyte

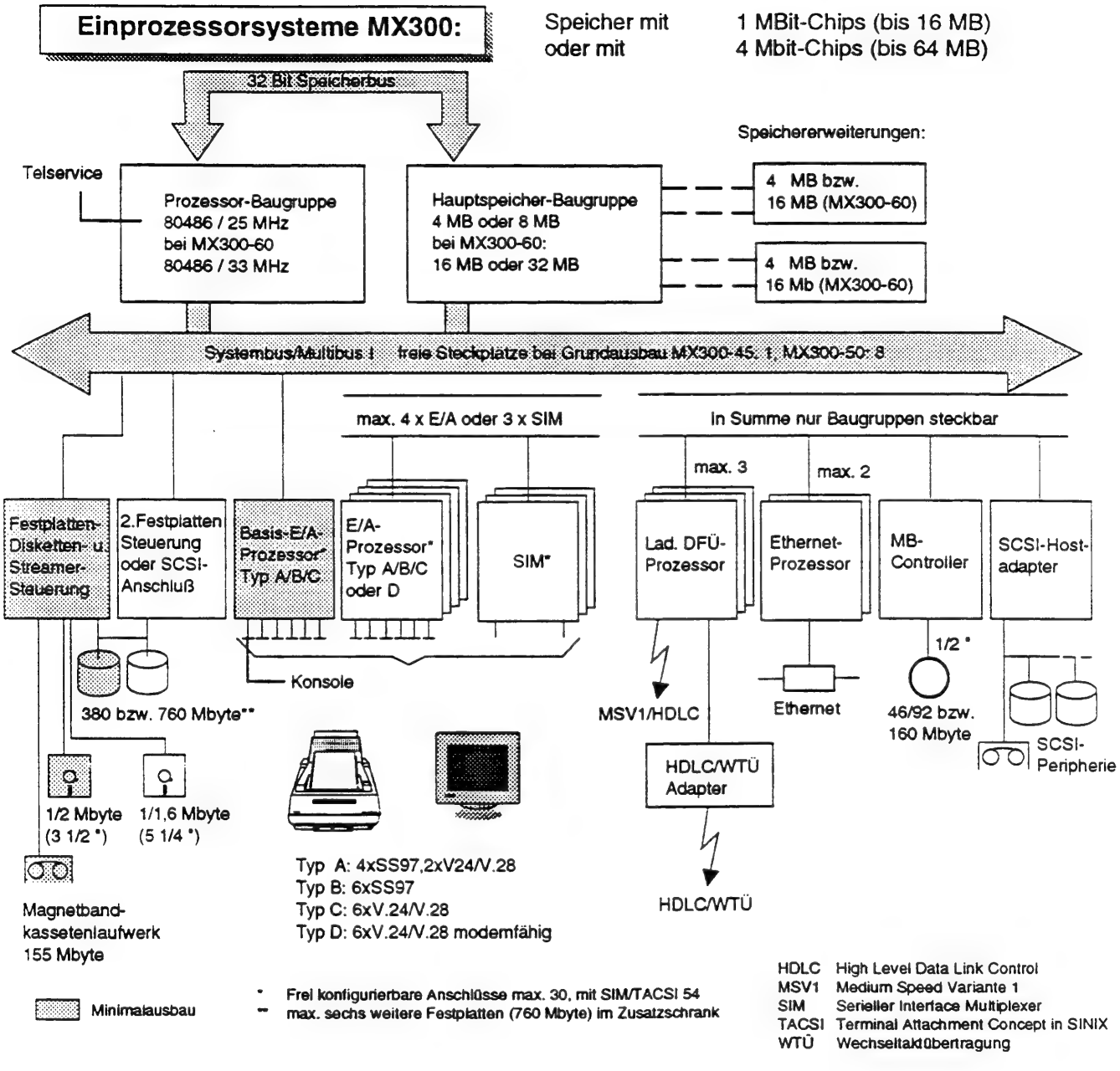
3 1/2" Floppy Disk

SINIX V5.40
- UNIX V.4 für Multiprozessorsystem

Umrüstmöglichkeit für MX500 Modell 75 auf Intel-Prozessoren

Auch Besitzer eines MX500-75 können die Intel 80486 Prozessoren nutzen. Der MX500-75 kann nachträglich mit Intel-Prozessoren und SINIX V5.40 ausgestattet werden. Damit steht auch diesen Anwendern die Welt von UNIX System V Rel. 4 offen. Eine komplette Hochrüstung auf alle Eckwerte des MX500-90 ist allerdings nicht möglich. Anstelle der bisher möglichen 4 Dualprozessorboards des MX500-75 können bis zu 4 Singleprozessorboards mit Intel 80486 eingesetzt werden. Alle anderen Maximalwerte des MX500-75 (HSP-Ausbau/ Festplatten-Kapazität / max. Schnittstellen) bleiben gleich.

Interner Aufbau der SINIX-Rechner



Festplattencontroller und Festplatten:

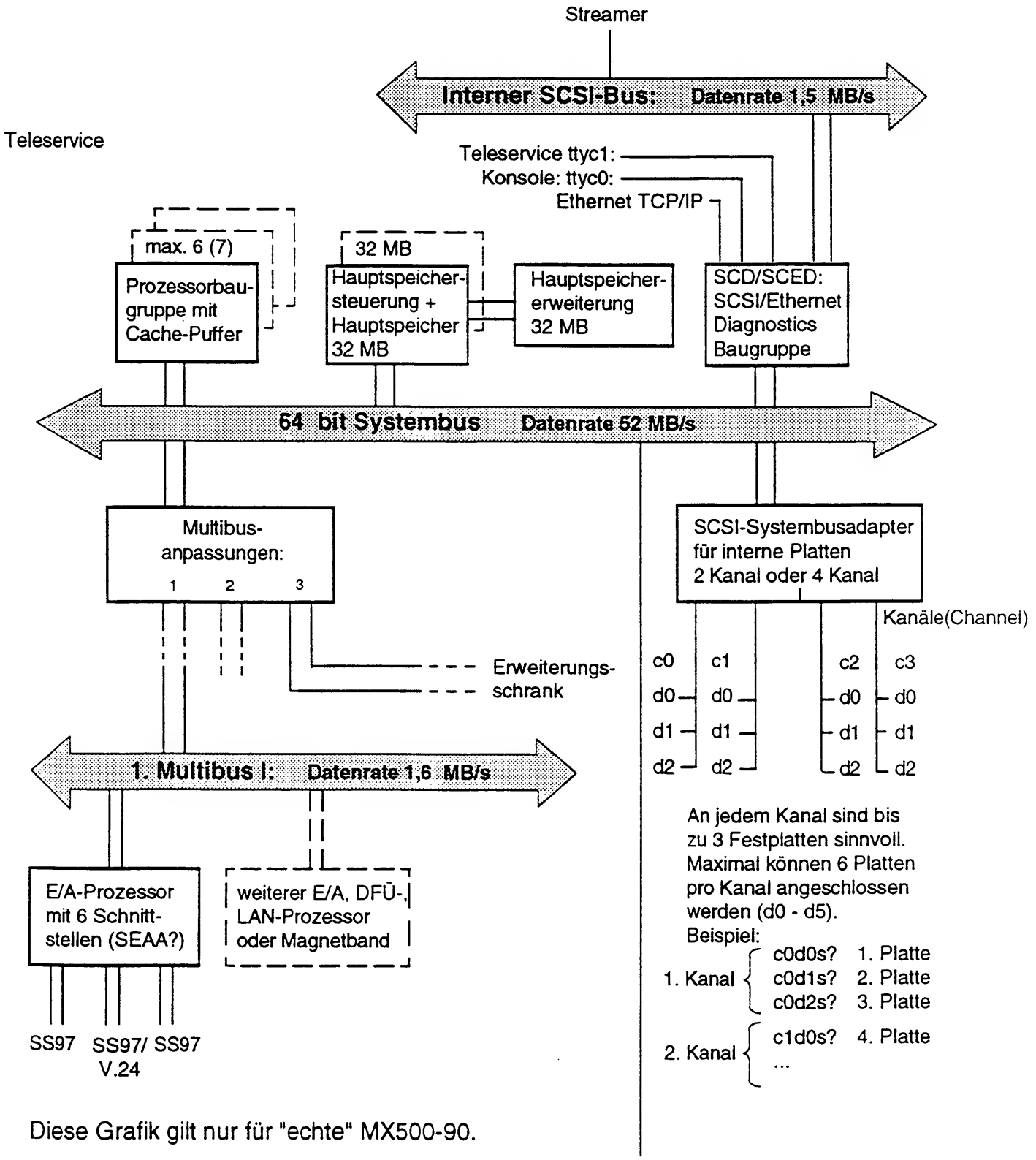
("Controller" == Plattensteuerung)

Bis zu 2 interne Platten mit ESDI-Steuerung:

- entweder beide Platten an der 1. Plattensteuerung
- oder 2. Platte an einer eigenen Plattensteuerung

Externe SCSI-Platten sind über eine spezielle SCSI-Steuerung anschließbar.

Parallelprozessorsystem MX500-90



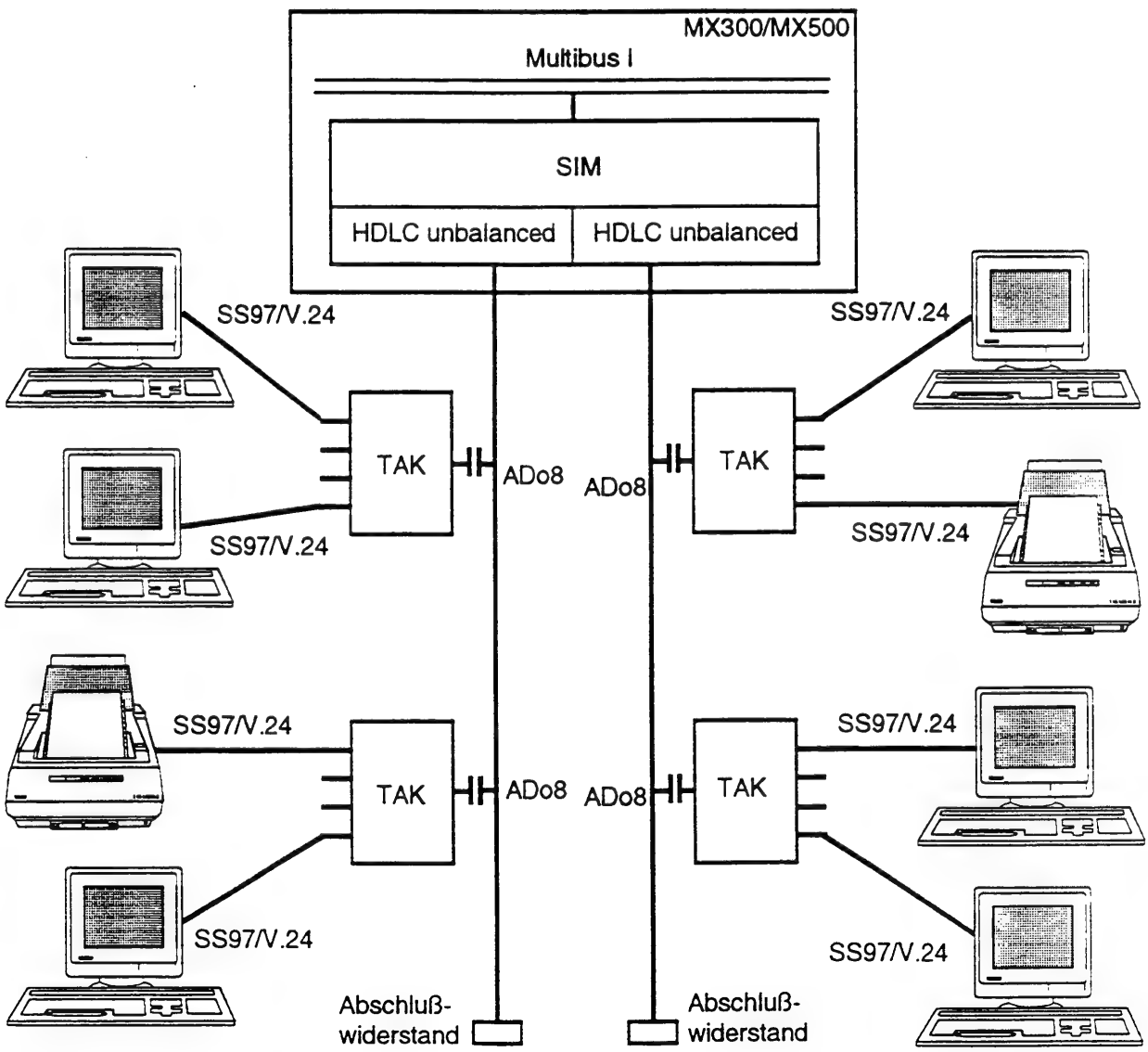
Diese Grafik gilt nur für "echte" MX500-90.

TACSI

TACSI (Terminal Anschluß Konzept in SINIX) verwendet anstelle der E/A-Prozessoren spezielle "SIM"-Multiplexer-Baugruppen (SIM == Serial Interface Multiplexer).

Jede SIM-Baugruppe unterstützt bis zu 16 Schnittstellen. Bis zu 8 Schnittstellen werden vom Multiplexer auf eine AFP-2-Draht-Leitung mit HDLC-Prozedur (187,5 Kbit/s) geleitet.

An jede AFP-2-Draht-Leitung können Sie 2 TAK-Geräte (TAK == Terminal Anschluß Konzentrator) mit maximal 4 Schnittstellen anschließen:

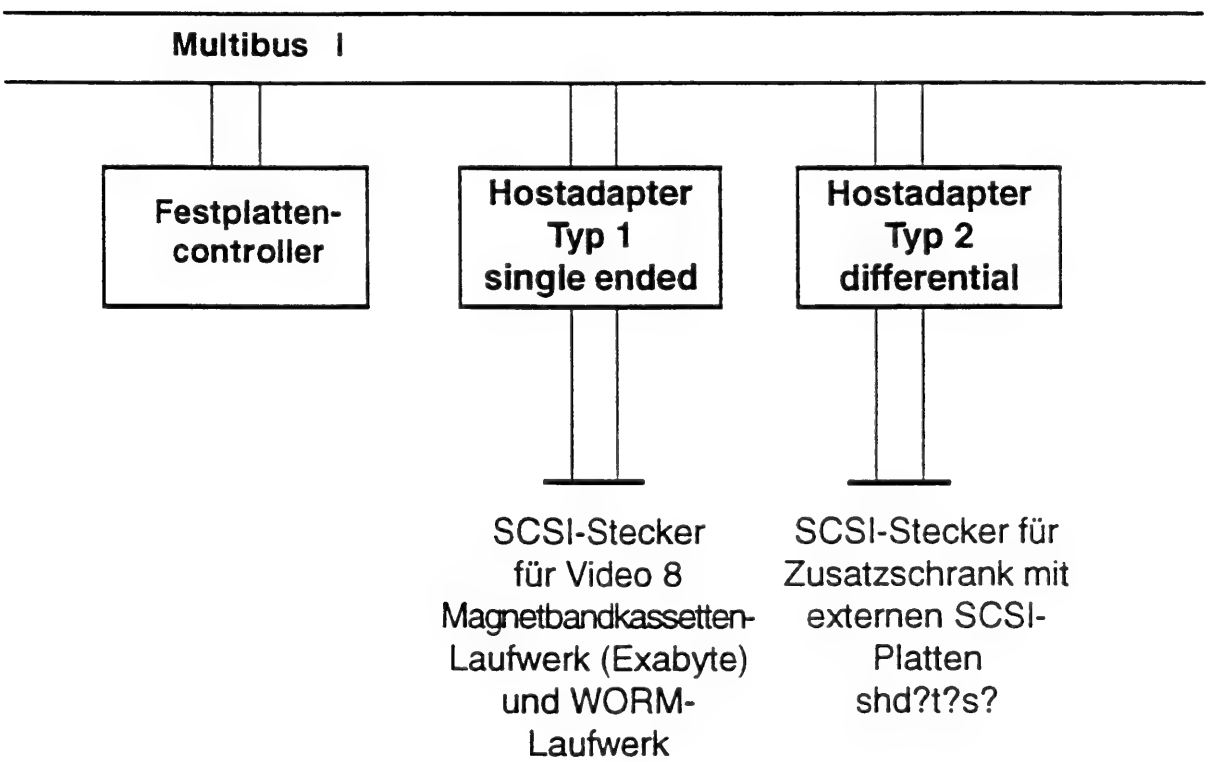


AFP Alternierendes Flanken-Puls-Verfahren
ADo8 Anschlußdose ADo8
TAK Terminal-Anschluß-Konzentrator
SIM Serial Interface Multiplexer für Multibus I
HDLC High-Level Data Link Control

Externe SCSI-Geräte

Anschluß externer Geräte mit SCSI-Schnittstellen über NCR "Hostadapter" (optional):

ncr0 ... bzw. adp 32.0
ncr1 ...



MX300: max. 2 Hostadapter, bei 2 Hostadaptern aber nur 1 Plattencontroller !

MX500: nur 1 Hostadapter pro Multibus

Benutzer und Benutzergruppen

Benutzerkennungen:

Der Systemverwalter sollte für jede Person, die am Rechner arbeiten darf, eine eigene Benutzerkennung einrichten.

Prinzipiell könnten mehrere Personen gleichzeitig unter der gleichen Benutzerkennung arbeiten.
Erfahrungsgemäß treten dabei jedoch meist Probleme oder Fehler auf.

⇨ pro Person eine eigene Benutzerkennung !!!

Welche Informationen speichert SINIX für jede Benutzerkennung

Benutzergruppen:

Beim Einrichten einer Benutzerkennung müssen Sie angeben, zu welcher Benutzergruppe der neue Benutzer gehören soll.

Bevor Sie eine Benutzerkennung einrichten, sollten Sie daher mindestens eine Benutzergruppe einrichten !!

Die Zugehörigkeit zu einer Benutzergruppe ist wichtig für die Zugriffsrechte auf Dateiverzeichnisse, Dateien und Programme.

Mit den Kommandos "**chmod**" und "**umask**" können Sie dafür sorgen, daß alle Benutzer einer Benutzergruppe auf gemeinsame Dateiverzeichnisse, Dateien und Programme zugreifen können.

Benutzerkennungen zur Systemverwaltung:

root Shellbenutzer mit unbeschränkten Rechten

⇨ ein falsches Kommando kann das gesamte System zerstören

Bestimmte Aufgaben sollte nur "root" erledigen:

- Kontrolle symbolischer Verweise
- System in den "run-level 1" herunterfahren
- Sicherung ganzer Plattenbereiche (Dateisysteme)
- Umkonfigurierung der Festplatte(n)
- Systemuhr stellen

admin Systemverwalter mit menügeführter Benutzeroberfläche (SINIX-Besonderheit):

⇨ Alpha-COLLAGE bzw. Grafik-COLLAGE

Aufgaben (Beispiele):

- Installation von Softwareprodukten
- Benutzergruppen und Benutzerkennungen definieren
- Kennwörter von Benutzern ändern
- Konfiguration von Bildschirmen und Druckern
- zusätzliche Druckergruppen definieren
- Postverteilerlisten verwalten
- LAN- bzw. WAN- Software konfigurieren

sysadm Systemverwalter mit AT&T Menü (englisch)

Was haben diese Benutzer gemeinsam? UID = 0

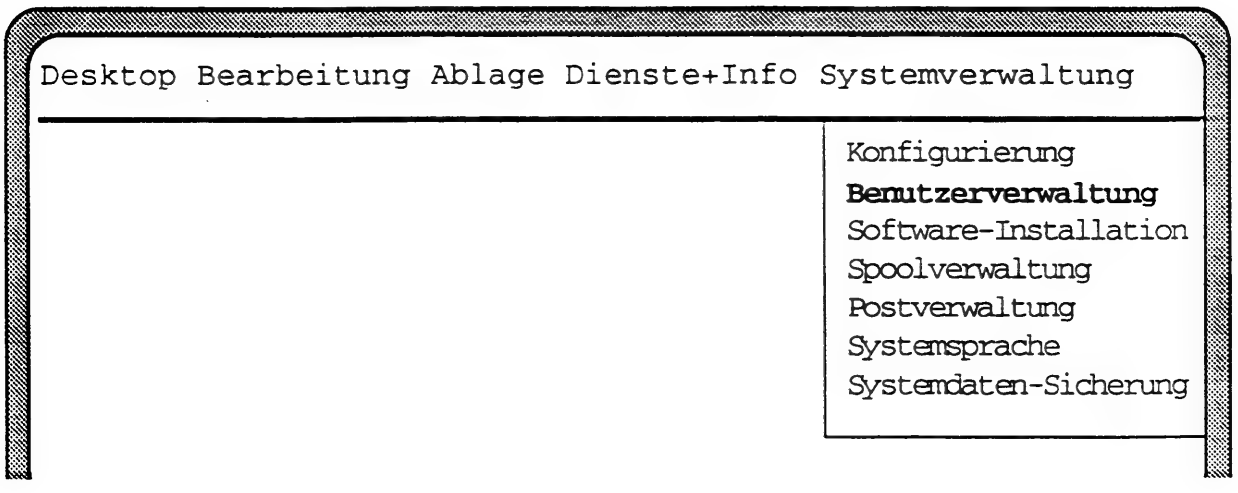
Menügeführte Systemverwaltung

⇨ Benutzerkennung: **admin**

COLLAGE

Drücken Sie die Taste

und verwenden Sie die Cursortasten.
Das Pull-Down-Menü "Systemverwaltung" :



Sie können das Menü nur auswählen, wenn ein Verzeichnisfenster offen ist und "oben" liegt.

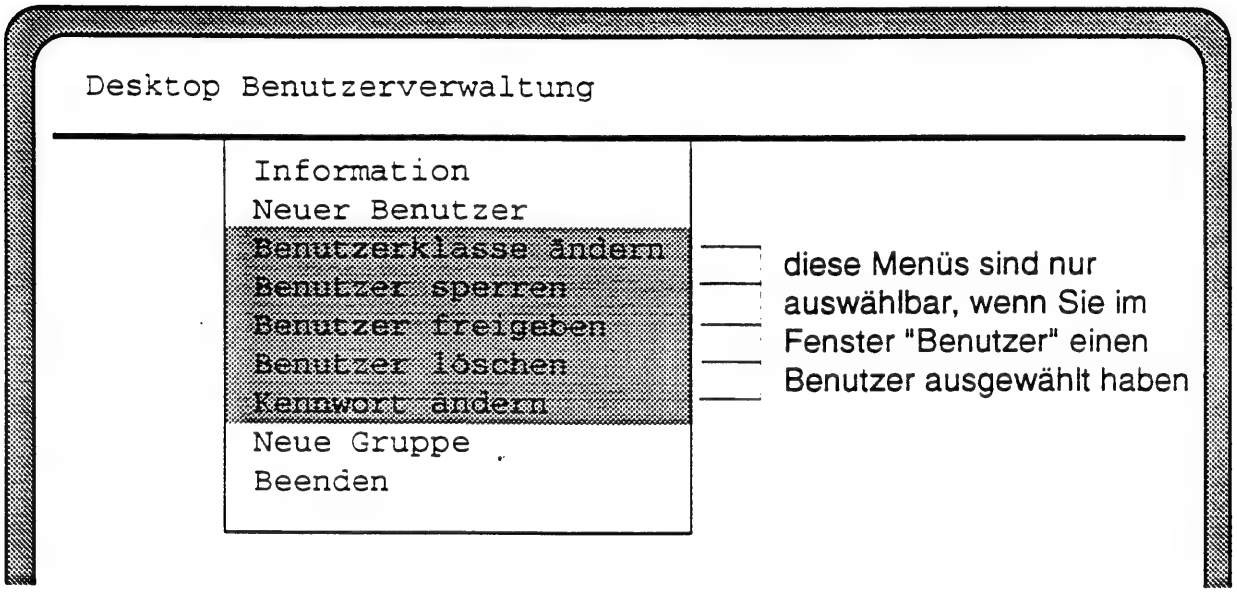
Benutzerverwaltung

Sie können Benutzergruppen und Benutzerkennungen über das "admin"-Bediensystem einrichten indem Sie aus dem Pull-Down-Menü "Systemverwaltung" auswählen (mit Cursortasten und):

Benutzerverwaltung

Sie starten damit ein spezielles Programm des Bediensystems. Dieses Programm dürfen Sie nicht von zwei Terminals gleichzeitig verwenden.

Sie erhalten ein Fenster "Benutzer", das eine Liste aller bereits eingetragenen Benutzer enthält. Drücken Sie wieder die Taste MENU . Sie haben eine neue Menüleiste mit folgendem Menü:



Sie können nun den Menüpunkt:

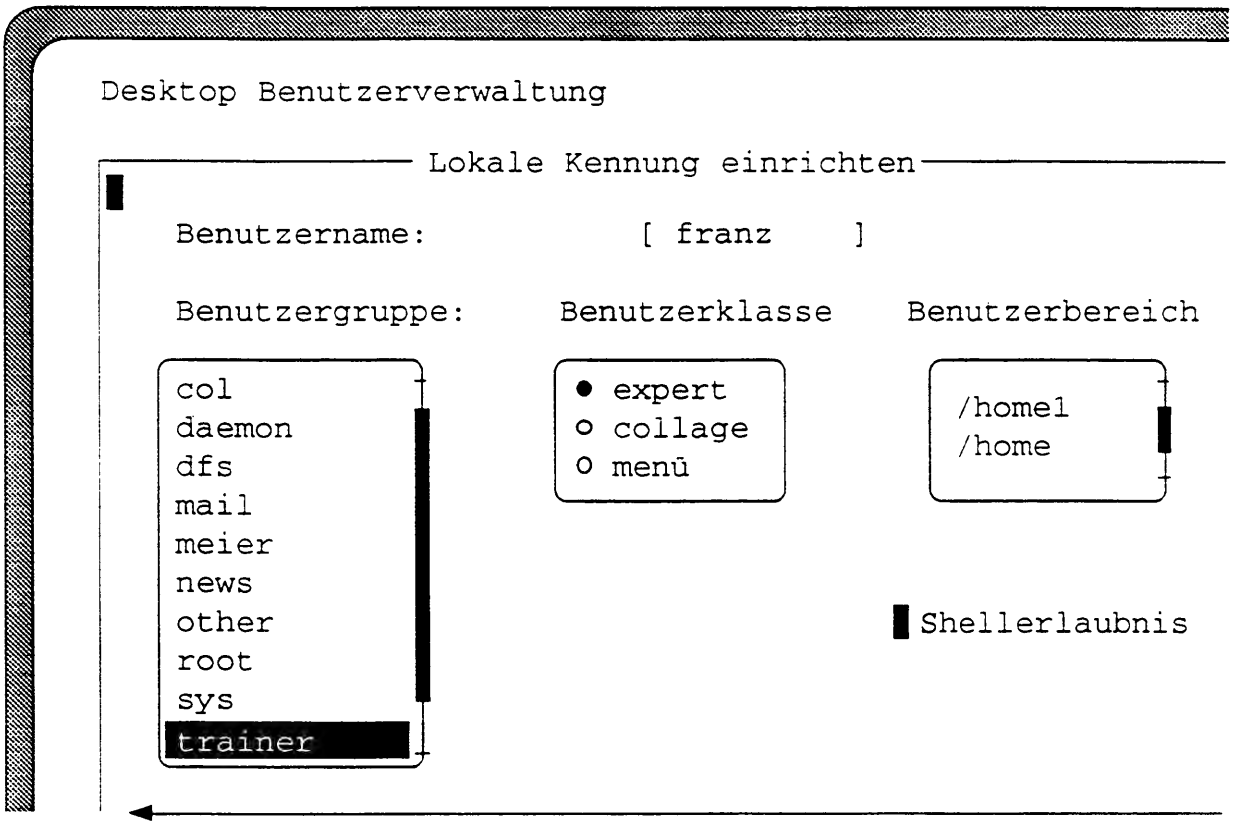
Neue Gruppe

auswählen.

Ist eine Benutzergruppe vorhanden, dann können Sie die Einrichtung eines neuen Benutzers beginnen:

Neuer Benutzer

Neuen Benutzer eintragen



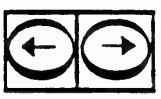
Mit den Tasten oder können Sie sich im Fenster bewegen.

Benutzerklasse:	- expert	Shellbenutzer
	- collage	COLLAGE-/bzw. OCIS-Benutzer
	- menü	altes Menüsystem
Desktops:	- collage	Benutzeroberfläche
	- ocis2	
	...	

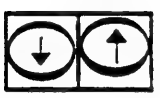
Benutzerbereich (Dateisystem): /home, /home1, ...

Wichtige Tasten von Alpha-COLLAGE

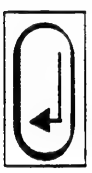
Pull-Down-Menüs herunterklappen



ein anderes Menü auswählen



Auswahl in einem Menü oder in einem Auswahlfenster



ein Objekt auswählen oder Auswahl an COLLAGE senden

zur nächsten Objektgruppe / Auswahl gehen
(wichtige Tasten für Alpa-COLLAGE)

die aktuelle Aktion abbrechen

ausgefülltes Fenster abschicken

Betriebsart

Betriebsart wechseln:
Alpha-COLLAGE besitzt 2 Betriebsarten
- Betriebsart "Anwendungen" (Standard)
- Betriebsart "System" (für Sonderfälle)

Sie erkennen die aktuelle Betriebsart an der Form des Cursors:

▮ Cursor für Betriebsart "Anwendungen"
In dieser Betriebsart werden fast alle Tasteneingaben an die aktive Anwendung weitergeleitet.

\+ Cursor für Betriebsart "System"
Der Cursor wird anstelle eines Mauszeigers verwendet.
Alle Funktions- und Sondertasten werden von Alpha-COLLAGE abgefangen

Wichtige Sondertasten:

gesamter Bildschirm ein/aus

aktives Fenster wiederherstellen

Alpha-COLLAGE Sondertasten

Betriebsart: "Anwendungen" (Standard)

In dieser Betriebsart werden fast alle Tasteneingaben die aktive Anwendung weitergeleitet.

Betriebsart: "System" (für Sonderfälle)

Der Cursor wird anstelle eines Mauszeigers verwendet.
Alle Funktions- und Sondertasten werden von Alpha-COLLAGE abgefangen.

Systemdateien zur Benutzerverwaltung: /etc/passwd, /etc/shadow

/etc/passwd: (lesbar für alle)

```
root:x:0:1:0000-Admin(0000):/:
daemon:x:1:1:0000-Admin(0000):/:
bin:x:2:2:0000-Admin(0000):/usr/bin:
            ...
sync:x:67:1:0000-Admin(0000):/:/usr/bin/sync
tele:x:90:1:0000-teleservice(0000):/home/tele:
install:x:101:1:Initial Login:/home/install:
nobody:x:100:101:unprivileged user:/nonexistent:/noshell
admin:x:0:2:Administrator mit COLLAGE:/usr/admin:/usr/bin/collage.pw
sysadm:x:0:0:AT&T administration:/usr/admin:/usr/sbin/sysadm
            ...
franz:x:104:103::/home/franz:/sbin/sh
meier:x:105:103:Rudi Meier:/home2/meier:/sbin/sh
```

Verweise auf /etc/shadow
Gruppennummer
Benutzer-nummer
wahlweise der vollständige Name
Login-Dateiverzeichnis
Startprogramm

/etc/shadow: (Verwaltung der Kennwörter)

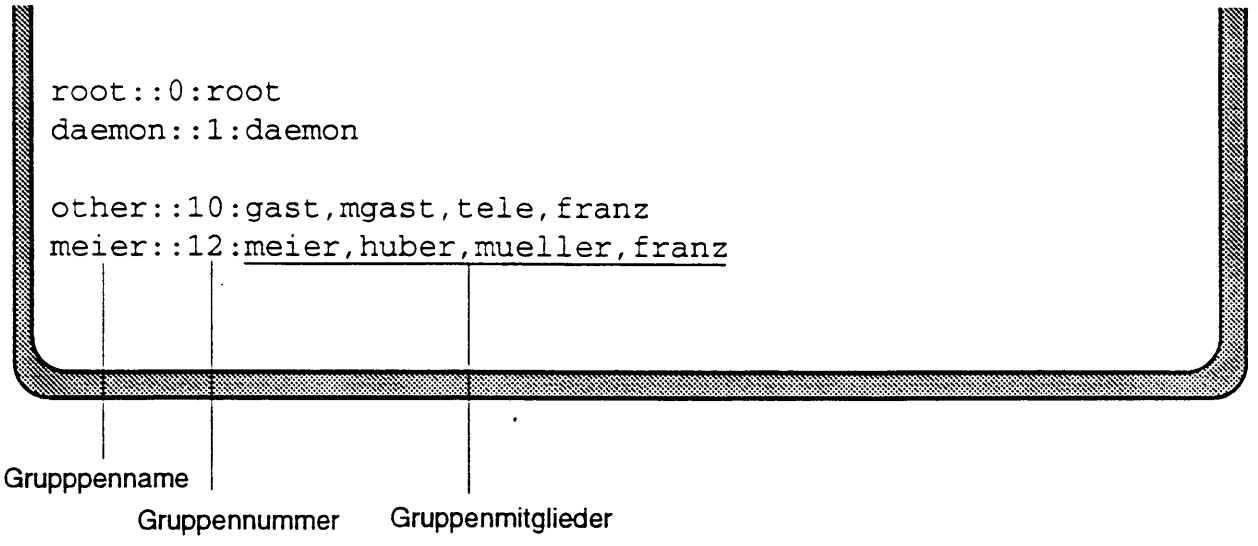
```
root:nn61ZTDxP0omo:7601:0:168:7:::
daemon:NONE:7601::::::
bin:NONE:7601::::::
            ...
sync::7601::::::
tele:NoCpAmPZiKyg2:7601:0:168:7:::
install:4o0Dfy3aYkECc:7601:0:168:7:::
nobody:*LK*:::::::
admin:2xno2kDQfcxkI:7602:0:168:7:::
sysadm:dtpaAwYdCc/sQ:7602:0:168:7:::
            ...
franz:NPkbOD/p/Xg9.:7624:0:168:7:::
meier:muNrPZEQx4xYs:7655:0:168:7:::
```

- verschlüsseltes Kennwort
- gesperrt: NONE, *LK*
- kein Kennwort (leeres Feld)

Tag der letzten Kennwort-Änderung (seit 01.01.1970)
Minimumzeit (in Tagen)
Maximumzeit
Warnzeit

Systemdateien zur Bentuzerverwaltung: /etc/group

/etc/group: Gruppenname <—> Gruppennummer (GID)



Sie können Benutzer zusätzlich als Mitglied in mehrere Benutzergruppen eintragen. Der Benutzer kann dann seine Gruppe wechseln mit dem Kommando:

```
newgrp Gruppe
```

Mit dem Kommando "groups" können Sie abfragen, zu welchen Gruppen Sie gehören.

Kommandos zur Benutzerverwaltung

```
                        # neue Benutzergruppe einrichten
groupadd GRUPPE

                        # Benutzergruppe löschen
groupdel GRUPPE
```

```
                        # neuen Benutzer einrichten
useradd  -g GRUPPE  -d /home/meier  -m  meier
```

Gruppe — HOME-Verzeichnis — HOME-Verzeichnis anlegen — Benutzer-name

Im Anschluß an "useradd" ist der Benutzer gesperrt. Sie müssen als nächstes ein Kennwort für den neuen Benutzer vergeben:

```
passwd meier
```

—> Kennwort eingeben

Weitere Schalter von "useradd":

`-s /usr/bin/ksh`	Startprogramm
`-c "Dr. Hans Meier"`	Kommentar

```
                        # Benutzer mit Home-DVZ löschen
userdel -r meier
```

Die Kommandos sind nicht verwendbar für Menü/(MES)-Benutzer !

Das "passwd"-Kommando

Mit dem "**passwd**"-Kommando können Sie Ihr eigenes Kennwort ändern.
Der Systemverwalter kann mit dem "**passwd**"- Kommando:

- Kennwörter und Verwaltungsinformationen beliebiger Benutzer ändern
- Verwaltungsinfomationen aller Benutzer ansehen

passwd -s benutzer

zeigt den Status des Kennworts und die Zeit-Angaben aus der "**/etc/shadow**"

Status: PS — Kennwort ist vorhanden
LK — Kennung gesperrt oder DFS Kennung
NP — kein Kennwort vorhanden

passwd -as # nur Systemverwalter

zeigt die obigen Angaben für alle Benutzer

passwd -l benutzer

sperrt die Benutzerkennung

passwd -x max_tage -n min_tage benutzer

legt die maximale und die minimale Gültigkeitsdauer in Tagen fest

opasswd und oshadow

Bei jeder Änderung in der "**passwd**" oder der "**shadow**"- Datei sichert das "**passwd**"-Kommando den alten Stand in der "**/etc/opasswd**" beziehungsweise "**/etc/oshadow**".

/etc/default/login, /etc/default/su

In der Datei "**/etc/default/login**" können Sie Parameter für jedes Login setzen:

Beispiel:
```
HZ=100
ULIMIT=32000
CONSOLE=/dev/term/tty000
PASSREQ=YES
ALTSHELL=YES
PATH=/usr/bin:/usr/sbin:/opt/bin:.
SUPATH=/usr/bin:/usr/sbin:/opt/bin:/sbin:/opt/etc:/usr/ucb
```

Bedeutung der Variablen:

CONSOLE	Falls dieser Parameter gesetzt ist, kann sich der "**super-user**" nur über das angegebene Terminal anmelden. Ist der Parameter nicht gesetzt, kann sich der "**super-user**" an jedem Terminal anmelden.
PASSREQ=YES	alle Benutzer müssen ein Passwort haben (bis auf "**root**")
ALTSHELL=YES	die Variable SHELL darf einen anderen Wert annehmen als "**/bin/sh**"
SUPATH	PATH für Logins mit der Benutzernummer 0 Dieser PATH gilt nicht für das "**su**"-Kommando.

In der Datei "**/etc/default/su**" können Sie Parameter für das "su"-Kommando setzen:

Beispiel:
```
SULOG=/var/adm/sulog
CONSOLE=/dev/term/tty000
SUPATH=/usr/bin:/usr/sbin:/opt/bin:/sbin:/opt/etc:/usr/ucb
```

CONSOLE	Falls dieser Parameter gesetzt ist, gibt das "**su**"-Kommando bei jedem "**root**"-su eine Meldung am angegebenen Terminal aus.
SUPATH	PATH nach "**root**"-su

Anmelden über X-Terminals:

Falls sich ein Benutzer über ein X-Terminal anmeldet, geschieht dies momentan weder über das login- noch über das rlogin-Programm, sondern meist über das X-Programm "xdm".

Der "xdm" berücksichtigt nicht die Datei /etc/default/login.
Die Path-Variable für Benutzer an X-Terminals können Sie z.B. setzen in der Datei: "/opt/lib/X11/xdm/Xsession.TX100" oder in der "/etc/profile".

Weitere Kommandos zur Systemverwaltung:

chmod Zugriffsrechte verändern

```
chmod ug+rw ....
chmod ug+x ....
```

chown Eigentümer verändern

```
chown meier ....
```

In SINIX V5.4 besitzen die Kommandos "chmod", "chown" und "chgrp" einen neuen Schalter "-R" (rekursiv). Dieser Schalter bewirkt, daß die Kommandos auch alle Unterverzeichnisse bearbeiten:
chown -R meier /home/meier

chgrp Gruppe verändern

```
chgrp gruppe4 ....
```

umask Zugriffsrechte für neue Dateien verbieten (vgl. /etc/profile)

```
umask 027          # Gruppenmitglieder dürfen nicht schreiben,
                   # andere dürfen weder lesen noch schreiben
```

Eigentümer — Gruppe — "andere"

1 == ausführen
2 == schreiben
4 == lesen

verbieten

#Jedes Bit, das Sie bei "umask" setzen,
ist bis zum Ende der aktuellen Shell für
#neuangelegte Dateien verboten.

du -s belegten Plattenplatz feststellen (in 512 Byte-Blöcken)

```
du -s /home/mueller
```

df -k Plattenbelegung des Gesamtsystems (in KB)

whodo, **finger** Informationen über Aktivitäten der angemeldeten Benutzer

Korn Shell: ksh

Mit SINIX V5.4 wird auch die "ksh" ausgeliefert. Diese Shell ist kompatibel zur normalen "Bourne Shell" ("sh"), bietet aber unter anderem die Möglichkeiten:

- die Kommandozeile mit "vi"-Kommandos zu editieren (wahlweise auch mit "emacs"-Kommandos)
- alte Kommandozeilen erneut zu bearbeiten (history)

Aufruf:

```
ksh -o vi          # Korn Shell mit "vi"-Kommandos
```

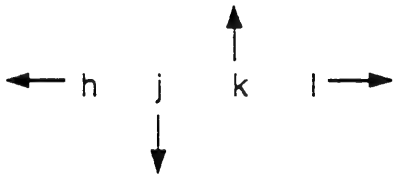
oder

```
ksh                # Korn Shell ohne Editor
set -o vi          # umschalten auf "vi"-Kommandos
```

In Editiermodus umschalten: <ESC>

Editiermodus verlassen: <ENTER> oder <DEL>

Es gelten die original "vi"-Cursor-Tasten:



Im Editiermodus können Sie alle original "vi"-Kommandos verwenden, nicht jedoch die Cursor-Tasten der Tastatur!

Die letzten Kommandozeilen ausgeben: `history`

Die "ksh" numeriert alle Kommandozeilen und speichert sie in der Datei ".sh_history".

Beispiele für history:

25. Kommandozeile editieren: `<ESC>25G`

Kommandozeile editieren,
die mit "prtvtoc" beginnt : `<ESC>/prtvtoc`

Kommandos, Tagesmeldung, Eröffungsbildschirm

wall Meldung an alle Terminals senden (Kommando für Systemverwalter)

"ps" -Kommando:

Liste aller Prozesse:	`ps -ef | pg`
alle eigenen Prozesse auflisten :	`ps -fu $USER`
alle Prozesse von Terminal tty005:	`ps -ft tty005`
Prozeßgrößen kontrollieren: (Spalte SZ in 4KB-Seiten)	`ps -el | pg`

kill PID Prozeß "sanft" beenden (Signal 15)

kill -1 PID Prozeß mit Signal 1 beenden (auch "sanft")

kill -9 PID Prozeß "brutal" beenden (Signal 9)

kill -9 0 alle Prozesse des eigenen Terminals brutal beenden (nicht bei COLLAGE)

Tagesmeldung, Eröffnungsbildschirm:

/etc/motd Datei mit Tagesmeldung (muß für alle lesbar sein und wird bei jedem Anmelden ausgegeben)

/etc/issue Eröffnungsbildschirm für alle Terminals (muß für alle lesbar sein)

t-Bit bei "/tmp" und "/var/tmp"

- Unter welchen Voraussetzungen kann ein normaler Benutzer fremde Dateien löschen ?

- Was bedeutet dies normalerweise für Dateien unter /tmp, /var/tmp und anderen allgemeinen Verzeichnissen ?

- Prüfen Sie die Zugriffsrechte von /tmp in SINIX V5.4.
  Versuchen Sie, fremde Dateien zu löschen!

- Probieren Sie aus:

```
cd $HOME
mkdir ALLE
chmod 777 ALLE
chmod +t ALLE
echo hallo > ALLE/datei
```

Bitten Sie nun Ihre Nachbargruppe, eine Datei "datei2" in Ihrem DVZ anzulegen und Ihre Datei zu löschen.

Aufgabe 1:

a) Sie übernehmen die Systemverwaltung eines vom Werk installierten SINIX-Rechners. Bei welchen Benutzerkennungen sollten Sie sofort die Kennwörter ändern ?
Wie ändern Sie die Kennwörter?

b) Ihr Kursleiter teilt Ihnen mit, wie Sie das COLLAGE-Bediensystem starten können:

```
acollage
```

Machen Sie sich mit der Bedienung dieser Benutzeroberfläche vertraut. Sie sollten jedoch nur Aktionen starten, die für alle Benutzer erlaubt sind.
Wählen Sie z.B. aus dem Menü "Dienste + Info" die Auswahl "Konfigurierung zeigen" und überprüfen Sie die aktuelle Konfigurierung von Terminals und Druckern.

Verwenden Sie für die folgenden Aufgaben die "ksh" und nutzen Sie die Editiermöglichkeiten.

c) Stellen Sie fest:

- wieviel Plattenplatz die Daten in Ihrem HOME-Verzeichnis belegen ("du"-Kommando)
- wieviel Platz auf der Platte noch frei ist
- mit welchen Kommandos/Programmen die anderen Kursteilnehmer im Moment arbeiten

Fortsetzung von Aufgabe 1:

d) Ändern Sie die Zugriffsrechte aller Dateien und Datei-Verzeichnisse unterhalb Ihres Login-Verzeichnisses:

Erweitern Sie die Zugriffsrechte für Ihre Gruppe und zwar:

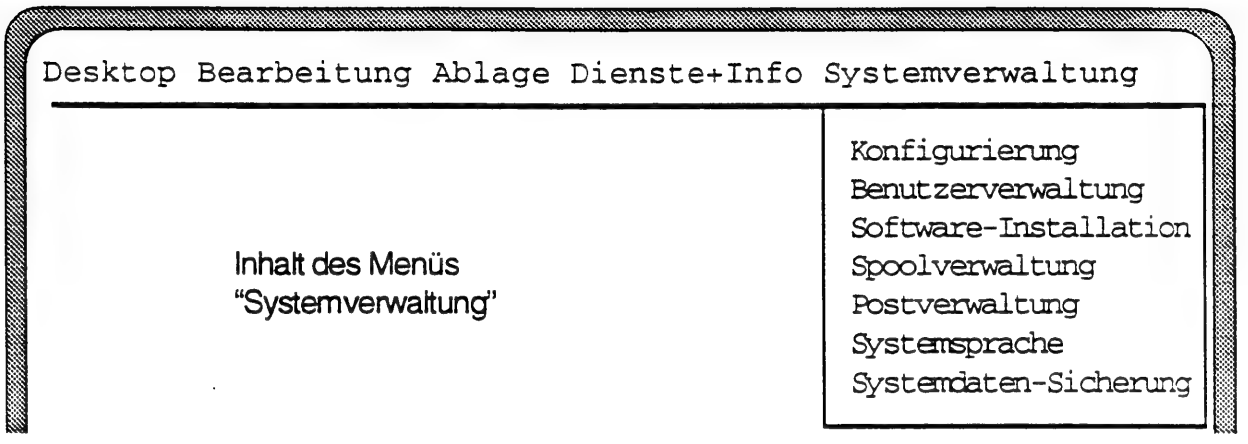
rw —	für Dateien
rwx —	für Dateiverzeichnisse

Überprüfen Sie das Ergebnis mit "ls -lR".

e) Sorgen Sie dafür, daß Ihre Gruppenmitglieder alle von Ihnen in Zukunft neu angelegten Dateien lesen können!

Konfigurierung der Peripherie

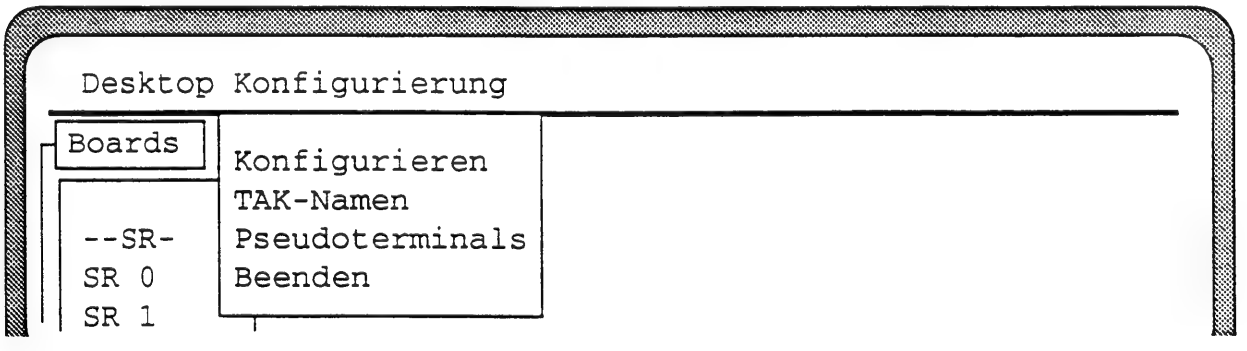
Die Konfigurierung der E/A-Prozessoren für Terminals, Drucker und anderer Peripheriegeräte sollten Sie ausschließlich über das "admin"-Bediensystem durchführen.



Sie können das Menü nur auswählen, wenn ein Verzeichnisfenster offen ist und "oben" liegt.

Über den Menüeintrag "Konfigurierung" starten Sie das Programm "**Konf**" des Bediensystems. Dieses Programm dürfen Sie nicht von zwei Terminals gleichzeitig verwenden.

Sie erhalten ein Fenster "Boards", das eine Liste aller vorhandenen E/A-Baugruppen enthält. Außerdem erhalten Sie eine neue Menüleiste mit folgendem Menü:

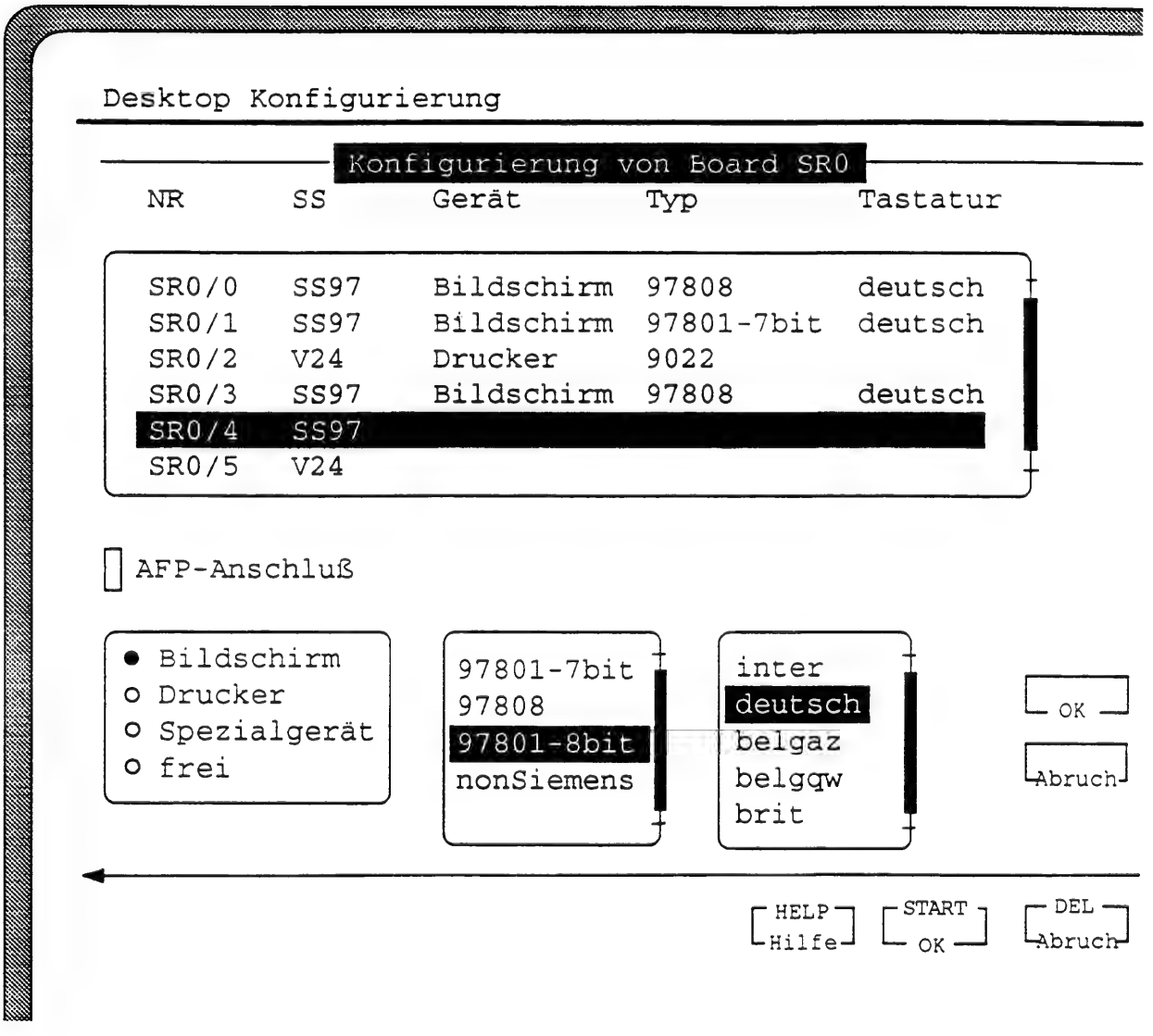


Sie wählen im Fenster "**Boards**" eine E/A-Baugruppe aus und anschließend aus dem Menü den Eintrag"**Konfigurieren**".

Das Bediensystem zeigt Ihnen die aktuelle Konfigurierung der betreffenden E/A-Baugruppe. Sie wählen eine oder mehrere Schnittstellen aus, die Sie umkonfigurieren wollen.

Auswahlmöglichkeiten:

Das Bediensystem gibt Ihnen nun zahlreiche Auswahlmöglichkeiten für diese Schnittstelle:



Haben Sie Ihre Auswahl für diese Schnittstelle beendet, so betätigen Sie die kleine OK-Marke.

Haben Sie Ihre Auswahl für die gesamte E/A-Baugruppe fertig, dann drücken Sie die START-Taste.

Systemdateien zur Terminalverwaltung

Systemdateien der verschiedenen UNIX/SINIX Versionen:

	UNIX V.40	SINIX 5.4/SINIX-ODT UNIX V.3	SINIX 5.2x Berkeley-UNIX
Konfigurationsdatei für Terminals	Kommando: "sacadm"	/etc/inittab	/etc/ttys
Programm zur Terminalverwaltung	ttymon	getty	getty
Schnittstellen-parameter	/etc/ttydefs Kommando: "sttydefs"	/etc/gettydefs	/etc/gettytab
TERM-Variable (Terminaltyp)	/etc/ttytype		
Terminal-initialisierung	-	/etc/termtab is=...:LK..=...	/etc/termcap is=...:LK=..
Eröffnungs-bildschirm	/etc/issue		/etc/herald/*

Die Datei /etc/ttytype:

```
97801 pts*                 ——— Pseudoterminals
97801 term/tty000          ——— Standardterminal
97808 term/tty003          ——— Grafik-COLLAGE-Terminal
```

Terminalname (Schnittstelle)

TERM-Variable

Die Datei /etc/termtab (nur SINIX V5.4):

```
term/tty000:is=\E[21u\08 \08\E(K\E[7u:LK=deut
term/tty003:is=\E[21u\08 \08\E(B\E[6u:LK=int
```

inter

Terminalname

Steuerzeichen zur Initialisierung

Tastaturtabelle für /sbin/keyload

gettydefs oder ttydefs

UNIX V.4 Systeme, die mit "getty"-Prozessen arbeiten (z.B. SINIX V5.4), benötigen die Datei "/etc/gettydefs":

Auszüge aus der /etc/gettydefs von SINIX V5.4:

```
S# B38400 OPOST ONLCR ... ISTRIP ... CS7 CREAD PARENB PARODD # B38400 OPOST ... PARODD #login: #S

N# B38400 OPOST ONLCR ... CS8 CREAD # B38400 OPOST ... CREAD #login: #N

B# B38400 OPOST ONLCR ... ISTRIP ... CS7 CREAD PARENB PARODD # B38400 OPOST ... PARODD  #login: #B

V# B19200 OPOST  ONLCR ... CS8 CREAD # B19200 OPOST ... CREAD # login: #v
```

Label der Zeile, die als nächstes verwendet wird

Treibereinstellungen (siehe "stty"-Kmdo)

ennzeichnung/Label
er Zeile (siehe/etc/
ittab)

Initialisierungs-einstellung

endgültigeEinstellung

Aufforderungstext zur Eingabe der Benutzerkennnung

UNIX V.4. Systeme, die mit "ttymon" arbeiten, benötigen die Datei "/etc/ttydefs":

Auszüge aus einer /etc/ttydefs:

```
S: 38400 opost  onlcr ... istrip ... cs7 cread parenb parodd : 38400 opost ... parodd : : S

N: 38400 opost  onlcr ... cs8 cread : 38400 opost ... cread : : N

B: 38400 opost  onlcr ... istrip ... cs7 cread parenb parodd : 38400 opost  ... parodd : : B

v: 19200 opost  onlcr ... cs8 cread : 19200 opost ... cread : : v

auto:hupl:ixany sane tab3 hupl:A:9600
```

"autobaud"

Die Datei "/etc/ttydefs" sollten Sie nur mit dem Kommando **"sttydefs"** bearbeiten.

terminfo statt termcap

In früheren UNIX-Versionen holten sich die Anwenderprogramme alle Steuerzeichen und Informationen zur Terminalsteuerung aus einer großen Datei "/etc/termcap".

Die neuen UNIX-Versionen stellen den Anwenderprogrammen kleine binäre "terminfo"-Beschreibungsdateien zur Verfügung.

In SINIX V5.4 wird die "/etc/termcap" nur noch für alte Anwenderprogramme benötigt, alle neuen Programme arbeiten unterhalb des Dateiverzeichnisses:

/usr/share/lib/terminfo

Die Beschreibungsdatei für das 97801-Terminal hat beispielsweise den Pfadnamen:

/usr/share/lib/terminfo/9/97801

Die binären Dateien können Sie mit dem Kommando **"infocmp"** in Quelldateien zurückverwandeln:

```
infocmp 97801 > 97801.ti
```

Sie können die Quelldatei verändern und mit dem **terminfo compiler (tic)** wieder neu übersetzen:

```
tic 97801.ti
```

"tic" legt das Ergebnis normalerweise wieder unter "/usr/share/lib/terminfo/?/..." ab.

TERMINFO

Über die Umgebungsvariable "TERMINFO" können Sie ein anderes Dateiverzeichnis für die terminfo Dateien einstellen. Falls diese Umgebungsvariable definiert ist, wird der Inhalt anstelle von "/usr/share/lib/terminfo" verwendet, um die binäre Datei zu finden oder zu erzeugen. Der Dateiname lautet dann:

$TERMINFO/?/ $TERM

Eigenschaften von "terminfo"

- Verbindliche Festlegung der Eigenschaften und ihrer Feldnamen (ähnlich ANSI X3.64-1979)
- Komma als Feldtrenner, Leerzeichen zwischen Feldern erlaubt
- Felder eines Eintrags sind durch führende Blank(s) oder Tabulator gekennzeichnet
- die leistungsfähige "curses"-Bibliothek für bildschirm-orientierte Anwendungen arbeitet mit "terminfo"
- Shellkommando zur Ausgabe von Steuerzeichen aus der terminfo-Datei:

```
tput clear     oder     US=`tput smul`
```

Feldnamen bei "terminfo":

terminfo	Bedeutung
cols/lines	Anz. Spalten/Zeilen des Bildschirms
cup	allgemeine Cursorpositionierung
cuf1/cub1	Cursor nach rechts / links
cud1/cuu1	Cursor nach unten / oben
el	Löschen: Cursor bis Zeilenende
ed	Löschen: Cursor bis Bildschirmende
clear	**Bildschirm löschen**
dl1/il1	Zeile löschen / einfügen
smcup	Terminalinitialisierung
smul/rmul	Unterstreichen ein / aus
kcud1 / kcuul1	Tasten "cursor down / up"
kcub1 / kcuf1	Tasten "cursor back / forward"
kich1 / kdch1	Tasten "INS CHAR / DEL CHAR
kf1 - kf20	Funktionstasten F1 - F20
kf23 - kf44	<Shift F1> bis <Shift F20>
lf0 - lf2	HELP, START, END
khlp, kress	HELP, START

Die SINIX Druckerverwaltung

lpr /opt/bin/lpr
Benutzerschnittstelle der Druckerverwaltung (wird von Benutzern oder von Programmen aufgerufen)

daemon /opt/etc/daemon
eigentliche Druckerverwaltung (2 Hintergrundprozesse)

/opt/etc/startup
Shellprozedur zum Start der Druckerverwaltung

/var/spool/spooler/tmp/daemtrc
Fehlermeldungsdatei

/var/spool/spooler/CONFIG

Textversion der Konfigurationsdatei:

Diese Datei enthält alle konfigurierten Drucker, Druckergruppen, Druckerverwalter und die Zuordnung von Terminals oder Benutzern zu bestimmten Druckergruppen.
Die CONFIG wird bei der Konfiguration vom "**admin**"- Bediensystem angelegt und kann vom Systemverwalter modifiziert werden.

/var/spool/spooler/FORMTAB

Das SINIX Spoolsystem erlaubt mehrere Warteschlangen je Drucker zur Unterstützung unterschiedlicher Papierformulare. In der FORMTAB Datei sind die erlaubten Formularnummern und Formularnamen hinterlegt.

/var/spool/spooler/sp/sp.*

Ruft der Anwender den "**lpr**" über Pipe oder mit einem der Schalter "**-cp**" oder "**+co**" auf, so kopiert der "**lpr**" die zu druckende Datei in dieses Dateiverzeichnis (temporäre Druckdatei).

Die CONFIG-Datei

1. Teil: konfigurierte Drucker

In jeder (langen) Zeile stehen folgende Informationen für einen Drucker (durch Blank getrennt):

- Druckername (max. 5 Zeichen)
- Backendaufruf mit absolutem Pfad (in Apostrophen)
- Absoluter Pfadname der Gerätedatei
- Liste aller zugelassenen druckerspez. lpr-Schalter

2. Teil: Definition der Druckergruppen

3. Teil: Liste der "Druckerverwalter"

4. Teil: Zuordnung von Terminals oder Benutzern zu Druckergruppen

Beispiel für eine CONFIG-Datei (mit verkürzten Zeilen):

```
D009 '/opt/etc/lp9001-b +hd' /dev/term/tty009 -zs= -zb= +tab -mar= -top= -za= -cat +cat -form=
D008 '/opt/etc/lp9001' /dev/term/tty008 -zs= -zb= -font= +tab -mar= -top= -za= -cat +cat -form=
D006 '/opt/etc/lp9004'/dev/term/tty006 -pl= -zs= -pb2 -hop= +tab -mar= -top= -za= -form=
D002 '/opt/etc/interface +lkmod -prog=/bin/cat +cbreak +odd -speed=B9600' /dev/term/tty002

9001 ( D008 D009 ) 'nur 9001 Drucker D008 D009 '
G009 ( D009 ) 'DRUCKER 9001-b'
G008 ( D008 ) 'DRUCKER 9001'
G006 ( D006 ) 'Hier muss ein Kommentar stehen !'
G002 ( D002 ) 'DRUCKER 9plott'

admin ( D009 D008 D006 D002 )
franz ( D009 )
meier ( D002 )

/dev/term/tty000 G009
/dev/term/tty001 G002
/dev/term/tty004 G008
/dev/term/tty007 G006
```

Die 4 Teile der CONFIG-Datei sind durch jeweils eine Leerzeile getrennt.
Zusätzliche Leerzeilen dürfen nicht vorkommen.

Neustart des Spoolsystems mit neuer CONFIG :

Wenn eine Veränderung der CONFIG-Datei wirksam werden soll, muß der Systemverwalter die Druckerverwaltung neu starten. Dies geschieht über das Kommando:

lpr -rr

Falls die Druckerverwaltung nicht mehr aktiv ist, können Sie sie starten mit der Startup-Prozedur:

/opt/etc/startup

Alle Meldungen, die die Druckerverwaltung beim Start ausgibt, stehen anschließend in der "daemtrc"-Datei :

/var/spool/spooler/tmp/daemtrc

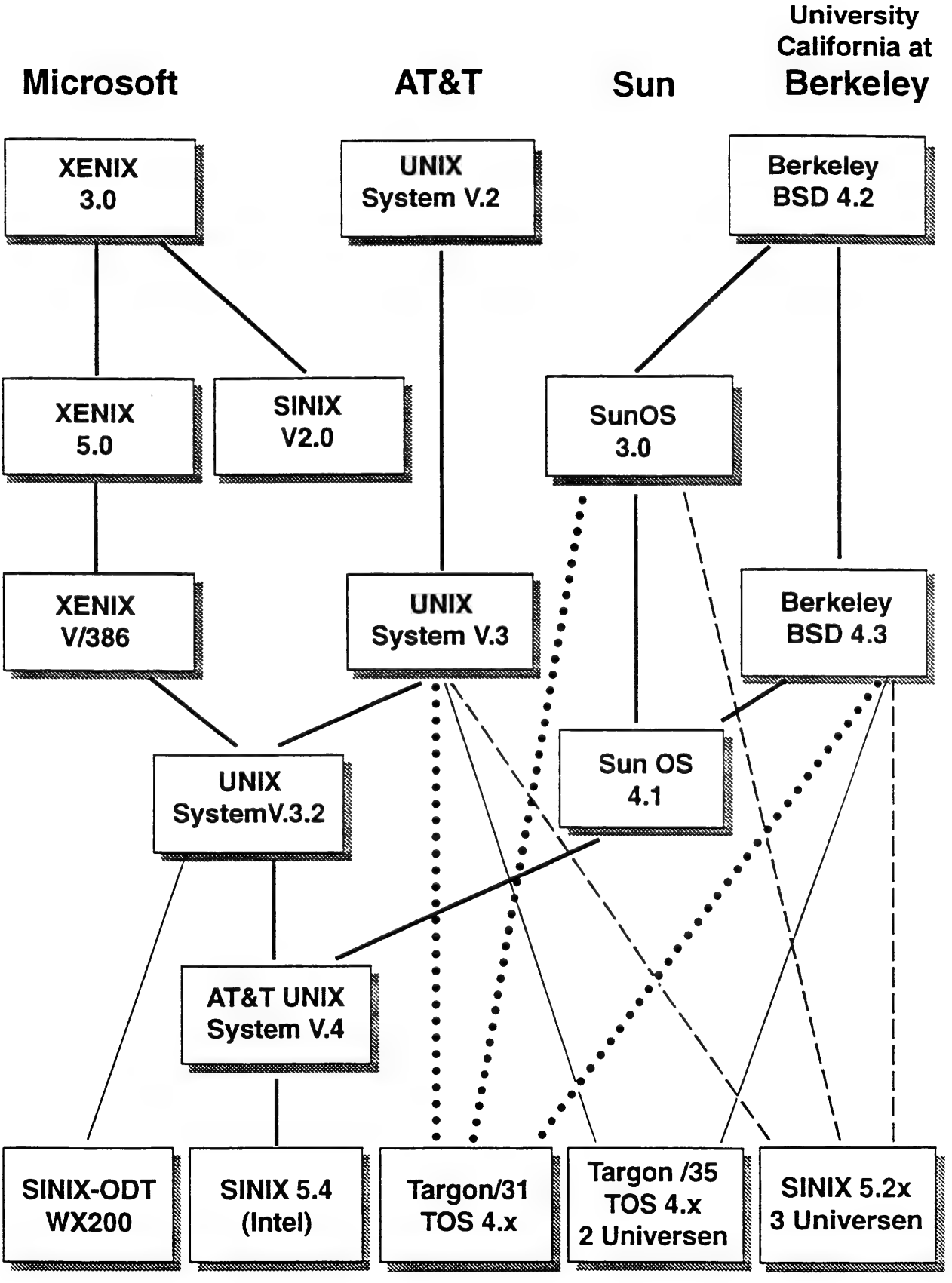
Falls die CONFIG-Datei Syntaxfehler enthält, stehen entsprechende Fehlermeldungen in der "**daemtrc**"-Datei.
In diesem Fall startet das Spoolsystem nicht und
alle Drucker sind "**UNBEKANNT**".

Es ist wichtig, daß alle Backends und Gerätedateien, die in der CONFIG-Datei angegeben sind, auch wirklich existieren und ansprechbar sind !
Andernfalls bleiben die betreffenden Drucker im Zustand"**UNBEKANNT**".

Legen Sie sicherheitshalber vor jeder Änderung der CONFIG ein Sicherheitskopie dieser Datei an !

Dateibaum, Plattenbereiche, Dateisysteme

UNIX/SINIX-Stammbaum



X/OPEN-Standards

Die X/OPEN-Standards für UNIX Systeme sind festgelegt im "XOPEN Portability Guide". Es gibt 2 "Portability Guides":

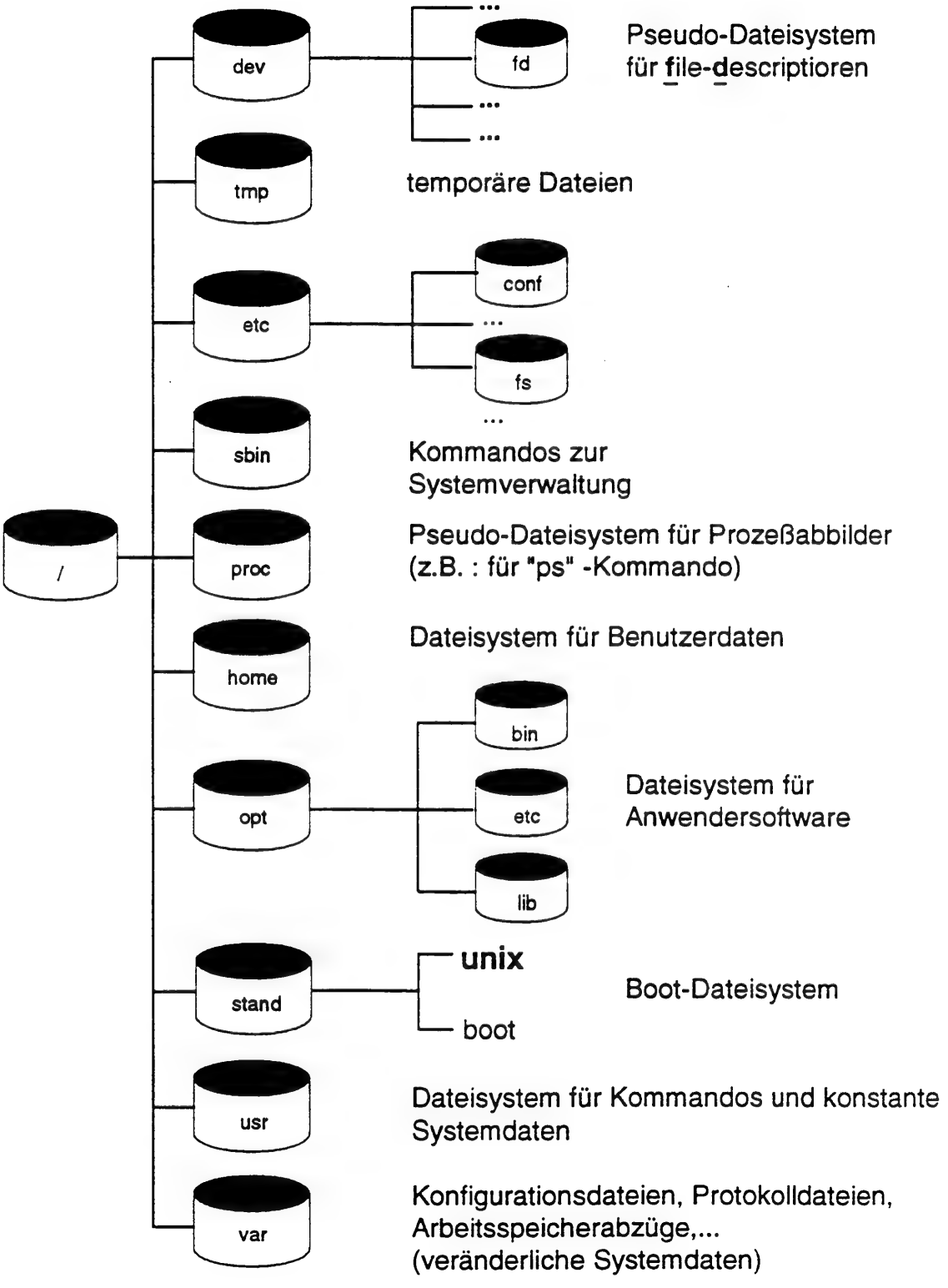
X/OPEN Portability Guide 2: XPG 2 (älterer Standard)

X/OPEN Portability Guide 3: XPG 3 (aktueller Standard)

SINIX V5.4 und SINIX V5.22/V5.23 entsprechen dem aktuellen X/OPEN Portability Guide 3.

Die Kommandos zur Systemverwaltung sind in den X/OPEN-Standards noch nicht festgelegt. Hier unterscheiden sich die SINIX Versionen 5.2x und 5.4 erheblich !

Dateibaum in SINIX V5.4/UNIX V.4



Aus Kompatibilitätsgründen wurden die alten Dateiverzeichnisse "**/bin**", "**/lib**", ... durch einfache symbolische Verweise auf die neuen Dateiverzeichnisse ersetzt.

Dateibaum, Plattenbereiche, Dateisysteme

Dateibaum in SINIX V5.4

/dev	Gerätedateien
/dev/fd	Pseudo-Dateisystem für file-descriptoren
/export	Montierpunkt für Dateisysteme, die von anderen Rechnern über ein Netz erreichbar sein sollen
/etc	Systemdateien, **keine** Kommandos
/etc/fs/...	Kommandos für verschiedene Dateisystemtypen
/etc/conf/..	Dateien, um einen Systemkern zu binden
/sbin	Kommandos und Programme zur Systemverwaltung
/home **/home1** **...**	eigene Dateisysteme für Benutzerverzeichnisse z.B: /home/meier, /home1/franz
/install	für Installation von Softwarepaketen
/opt	Dateisystem für Anwendersoftware
/proc	Pseudo-Dateisystem für Prozeßabbilder (kein Platzbedarf)
/shlib	aus Kompatibilität zu UNIX System V.3
/stand	Boot-Dateisystem mit Systemkern (ca. 5 MB)
/tmp	Dateiverzeichnis/Dateisystem für temporäre Dateien
/usr	Kommandos und konstante Systemdaten (/usr/bin, /usr/sbin, /usr/ucb, /usr/ccs, ...)
/var	Dateiverzeichnis/Dateisystem für Dateien, die laufend geändert werden (Konfigurations-,Protokolldateien, Arbeitsspeicherabzüge, ...)

Das "/var"-Dateisystem

/var/adm/	Accounting, Protokolldateien lastlog, sulog, utmp*, wtmp*
/var/adm/klog.msg	Meldungen des Systemkerns
/var/crash/	**Arbeitsspeicherabzüge des Systems**
/var/sadm/	Installationsprotokolle, eventuell Backup-Protokolle
/var/ced/	Hilfsdateien für den ced-Editor
/var/mail/	Verzeichnis für Postkörbe
/var/cron/log	Protokolldatei des "cron"
/var/spool/cron/	Dateien und Dateiverzeichnisse für "crontab" und für "at"-Kommando
/var/spool/spooler/	für SINIX-Spoolsystem
/var/tmp/	temporäre Dateien (früher: "/usr/tmp")
/var/saf/	Protokolldateien für Netzsoftware
/var/yp/	für "yellow pages" (Netzsoftware)

Das "/opt"-Dateisystem (40 - 100 MB)

Hier soll alle Anwendersoftware (optionale Software) installiert werden. Bei SINIX V5.4 wird das SINIX-Spoolsystem ("lpr") unterhalb von "/opt" installiert.

/opt/bin/	Programme, Kommandos z.B.: lpr, maxed
/opt/etc/	z.B.: interne Programme des SINIX-Spools
/opt/lib/ /opt/include/	Hilfsdateien/Dateiverzeichnisse
/opt/emds/ /opt/ft/	produktspezifische Verzeichnisse

Das "/usr"-Dateisystem (60 - 80 MB)

Hier sollten alle Systemprogramme und Dateien stehen, die sich nicht ändern.
Leider wird dieses Konzept noch nicht konsequent verfolgt.

/usr/admin/

/usr/bin/	**fast alle Kommandos**
/usr/sbin/	**zusätzliche Kommandos zur Systemverwaltung**
/usr/ccs/	original C-Entwicklungssystem von AT&T: normale Bibliotheken: lib*.a "shared libraries": lib*.so
/usr/ces/	SINIX C-Compiler (verwendet Bibliotheken aus "/usr/ccs/lib")
/usr/include/	
/usr/lib/	Hilfsdateien, normale Bibliotheken: lib*.a "shared libraries": lib*.so und : libc.so.1
/usr/share/	Dateien, auf die mehrere Rechner im Netz zugreifen dürfen (z.B. ONLINE-Manuale)
/usr/share/lib/terminfo	Terminfo-Verzeichnisse und Dateien
/usr/sadm/	für "sysadm", **nicht verwechseln mit "/var/sadm"**

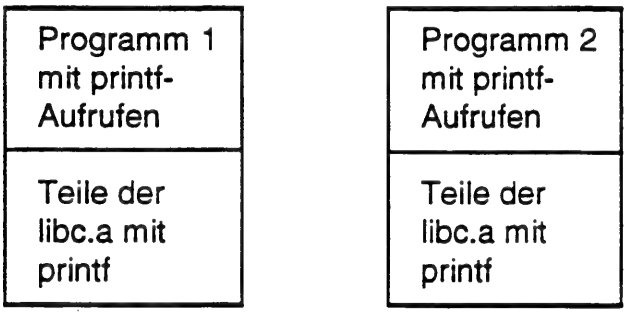
Falls Sie bei der Installation des Betriebssystems das Paket **"compat"** installiert haben, existieren die folgenden "Berkeley" Verzeichnisse:

/usr/ucb/	eine Auswahl von Berkeley-Kommandos z.B.: "mt", "ucb-ps", ...
/usr/ucbinclude/ /usr/ucblib/	C-Entwicklungsumgebung von Berkeley

Shared Libraries in V5.4

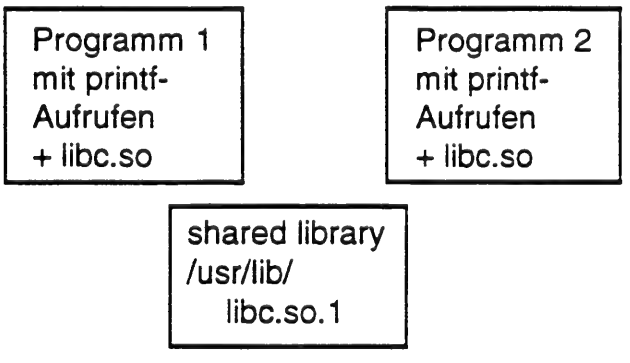
Statisch übersetzte Programme:

```
cc -o programm1 -dn programm1.c
cc -o programm2 -dn programm2.c
```



Programme, die dynamische "shared libraries" nutzen:

```
cc -o programm1 programm1.c
cc -o programm2 programm2.c
```



Die Programme werden gebunden mit "/usr/ccs/lib/libc.so" und sind nur lauffähig, wenn Sie zur Laufzeit auf "/usr/lib/libc.so.1" zugreifen können (Zugriff: r-x !).

Mit dem Kommando **ldd** können Sie prüfen, welche shared libraries zur Laufzeit von einem Programm benötigt werden:

```
ldd /usr/bin/ced

ldd /sbin/fsck

ldd /usr/bin/X11/mwm          # falls Motif installiert ist
```

Einfache symbolische Verweise:

Ein einfacher symbolischer Verweis ("symbolic link") ist eine spezielle Datei.
Sie enthält den Pfadnamen einer anderen Datei oder eines Dateiverzeichnisses.

Beispiele:

/bin ———> /usr/bin
/lib ———> /usr/lib

Aufgabe 2:

In SINIX V5.4 sind die meisten Kommandos unter anderen Pfadnamen erreichbar als in früheren SINIX / UNIX Versionen.

Aus Kompatibilitätsgründen wurden symbolische Links eingerichtet, so daß die Kommandos scheinbar unter den früheren Pfadnamen ansprechbar sind.

Prüfen Sie die Pfadnamen folgender Kommandos:

- wo stehen die Kommandos in SINIX V5.4 ?
- sind die Kommandos über symbolische Links auch unter anderen Pfadnamen erreichbar ?

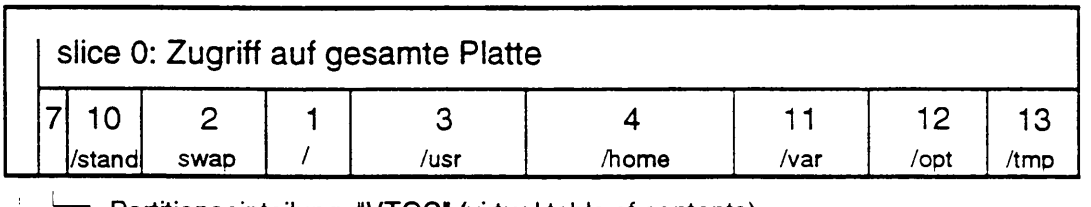
Kommando/ Programm	tatsächlicher Pfadname:	symbolische Verweise
sh		
ls		
tar		
ps		
passwd		

Dateibaum, Plattenbereiche, Dateisysteme

Logische Platteneinteilung

Eine Festplatte ist unterteilt in bis zu 15 Bereiche, die SINIX als getrennte "logische Plattenlaufwerke" behandelt.
Bezeichnung: Plattenbereich oder "slice" 0 bis 15

Beispiel (MX3000):



└— Partitionseinteilung: "VTOC" (virtual table of contents)

Zylinder 0: "boot"-Programm (Kopie von "/etc/boot")
und "fdisk"- Tabelle

Die Größe der einzelnen Bereiche oder "slices" können Sie bei einer Neu-Installation des Betriebssystems festlegen.

Aktuelle Platteneinteilung lesen:

Die Einteilung einer Partition steht im "VTOC".
Der Systemverwalter kann diese Tabelle über die Gerätedatei für "slice 0" lesen:

```
prtvtoc -f vtoc1  /dev/rdsk/c0d0s0
cat vtoc1         # 1.Platte

prtvtoc -f vtoc2  /dev/rdsk/c0d1s0 ||
                  prtvtoc -f vtoc2 /dev/rdsk/c1d0s0

cat vtoc2         # 2.Platte
```

Die Datei /etc/partitions

Die Platteneinteilung aller Platten sollte zusätzlich in der Datei "/etc/partitions" hinterlegt sein. Diese Datei wird jedoch nicht automatisch aktualisiert, so daß Sie sich auf den Inhalt nicht verlassen können (siehe "prtvtoc -e").

Dateibaum, Plattenbereiche, Dateisysteme

Sie können die Bereiche der Festplatte(n) (die "logischen Plattenlaufwerke") für verschiedene Zwecke nutzen:

- **"slice"** 2 der 1.Platte ist automatisch **"swap"**-Bereich
- In anderen Bereichen erzeugen Sie normalerweise **Dateisysteme,** die Sie in Ihren Dateibaum einhängen
- Falls Sie Turbo-Informix bzw. Informix-Online verwenden, reservieren Sie weitere Plattenbereiche exklusiv für Ihre Datenbanken und konfigurieren dort kein Dateisystem.
- beim MX500-90 reservieren Sie zusätzlich einen Slice "/dev/dump" für einen Speicherabzug

Die swap-Bereiche:

- Auslagerungsbereiche (für **"paging"** und **"swapping"**), die der Betriebssystemkern unbedingt benötigt
- Sie können mehrere **"swap"**-Bereiche konfigurieren
- **ist die Summe der "swap"-Bereiche zu klein, bleibt der Rechner bei Belastung "stehen"**

Gesamtgröße der "swap"-Bereiche:

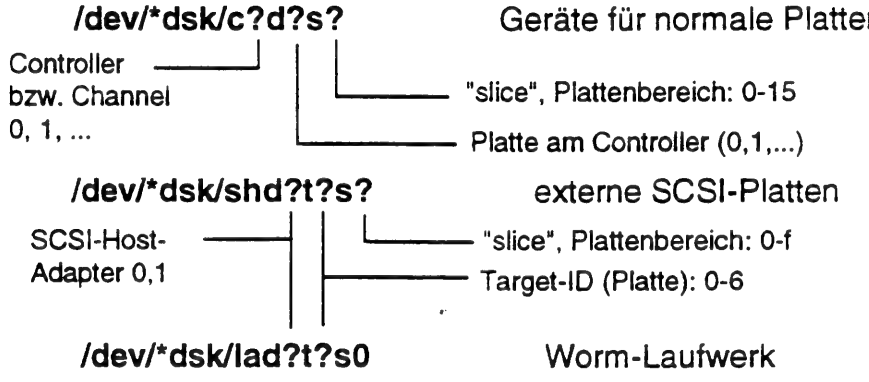
SINIX V5.4 : **mindestens 2 mal Hauptspeicherausbau !**

Dateibaum, Plattenbereiche, Dateisysteme

Das Verzeichnis /dev

In SINIX/UNIX 5.4 sind viele Gerätedateien in Unterverzeichnissen von /dev zusammengefaßt:

/dev/dsk/	"block"-Geräte für Platten und Disketten
/dev/rdsk/	"raw"-Geräte für Platten und Disketten



/dev/VP/	Pseudo-Geräte für Spiegelplatten-System
/dev/rmt/c0s0	Streamer (rewind), rmt == "raw magnetic tape"
c0s0n	Streamer (no rewind)
/dev/term/tty???	Terminals und Drucker SR-Baugruppe : tty000 - tty095 SIM-Baugruppe (TACSI) : tty200 - tty455 SX-Baugruppe (Modem) : tty500 - tty595
/dev/sim/	spezielle Geräte für TACSI
/dev/col/	Pseudo-Geräte für Grafik-COLLAGE
/dev/pts/	Pseudo-Geräte für Alpha-COLLAGE und LAN
/dev/dlpi/exos	Geräte für EXOS (LAN)
/dev/inet/	Geräte für Netzanschlüsse

Aus Kompatibilitätsgründen bestehen bei vielen Gerätedateien Verweise auf zusätzliche Namen:

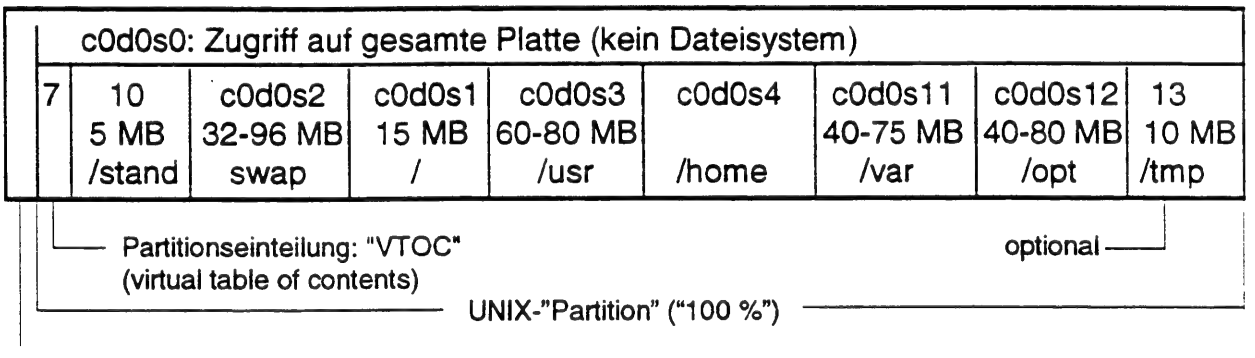
Beispiele: /dev/rmt/c0s0 === /dev/tape
/dev/pts/0 === /dev/pts000

Dateibaum, Plattenbereiche, Dateisysteme

Gerätedateien für Plattenbereiche:

Die Namen der Gerätedateien hängen davon ab, an welchem Channel bzw. Controller und an welchem Anschluß dieses Controllers die betreffende Platte angeschlossen ist.

Standardeinteilung bei einer "380 MB"-Platte:



Zylinder 0: "boot"-Programm (Kopie von "/etc/boot")
und "fdisk"-Tabelle

Gerätedateien für 2. Platte:

- Ist die 2. Platte am gleichen Controller angeschlossen wie die 1. Platte, so heißen die Gerätedateien:

  c0d1s0

- Ist die 2. Platte an einem eigenen Controller angeschlossen, heißen die Gerätedateien:

  c1d0s0

Plattentypen und Plattendaten

Plattentypen beim MX300:

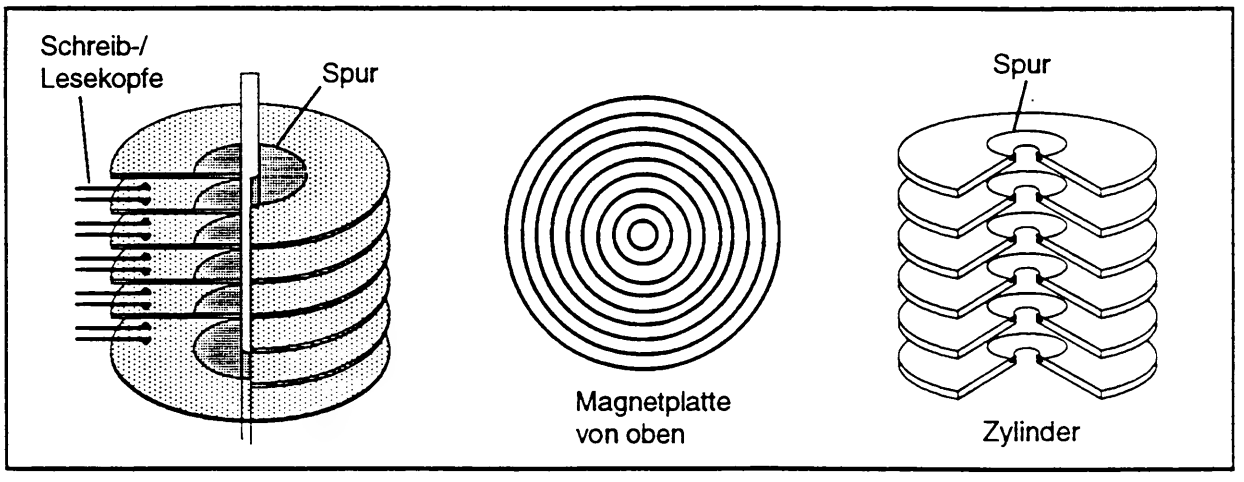
- "170 MB": MC1355, formatiert 139 MB (nur MX300-45)
- "300 MB": MegaFile/ME1300, formatiert 248 MB oder MC0558 (alt)
- "380 MB": MC1558, formatiert 310 MB oder MC1664, formatiert 322 MB
- "760 MB": MC1568 oder wren6, formatiert 649 MB oder ...

Plattendaten:

	"170 MB" MC1355	"380 MB" MC1558	"760 MB" MC1568	"380 MB" ... MC1664
Plattenköpfe: heads bzw. ntrack	8	15	15	7
Sektoren pro Spur —> nsect	34	34	53	53
ntrack*nsect == Sektoren/Zylinder	272	510	795	371
	(ein Sektor == 512 Bytes)			
==> KB / Zylinder	136	255	397	185

Plattendaten ausgeben lassen:

prtvtoc -p /dev/rdsk/c?d?s0



Dateibaum, Plattenbereiche, Dateisysteme

Standardeinteilung der Festplatten beim MX300:

Plattentyp/ Hsp.-Ausbau	MC1355 4-8 MB	MC1558 4-16 MB	MC1558 32-64 MB	MC1664 4-16 MB	MC1664 32-64 MB	MC1568 8-16 MB	MC1568 32-64 MB
Brutokap.	170 MB	380 MB	380 MB	380 MB	380 MB	760 MB	760 MB
Nettokap.	136 MB	303	303 MB	322 MB	322MB	632 MB	632 MB
Zylinder a*)	1017	1217	1217	1773	1773	1625	1625
/stand b*)	5(38)	5(21)	5(21)	5(28)	5(28)	5(12)	5(12)
/dev/swap b*)	16(122)	32(129)	96(387)	32(175)	96(530)	32(82)	128(330)
/ b*)	15(110)	15(60)	15(60)	15(28)	15(85)	15(38)	15(38)
/usr b*)	43(320)	60(241)	60(241)	60(330)	60(330)	60(155)	60(155)
/home b*)	27(207)	101(405)	37(147)	120(660)	55(305)	355(915)	252(649)
/var b*)	15(110)	40(160)	40(160)	40(220)	40(220)	75(192)	82(210)
/opt b*)	15(110)	40(160)	40(160)	40(220)	40(220)	70(180)	70(180)
/tmp b*)	---------	10(419)	10(41)	10(55)	10(55)	20(51)	20(51)

a*) : Es stehen effektiv 7 Zylinder mehr zur Verfügung, die jedoch reserviert sind.
b*) : Der erste Wert kennzeichnet die Slicegröße in MB (vor Generierung eines Filesystems).
Der Wert in Klammern stellt die korrespondierende Zylinderzahl des Slice dar.

Dateibaum, Plattenbereiche, Dateisysteme

System-Ressourcen beim MX300:

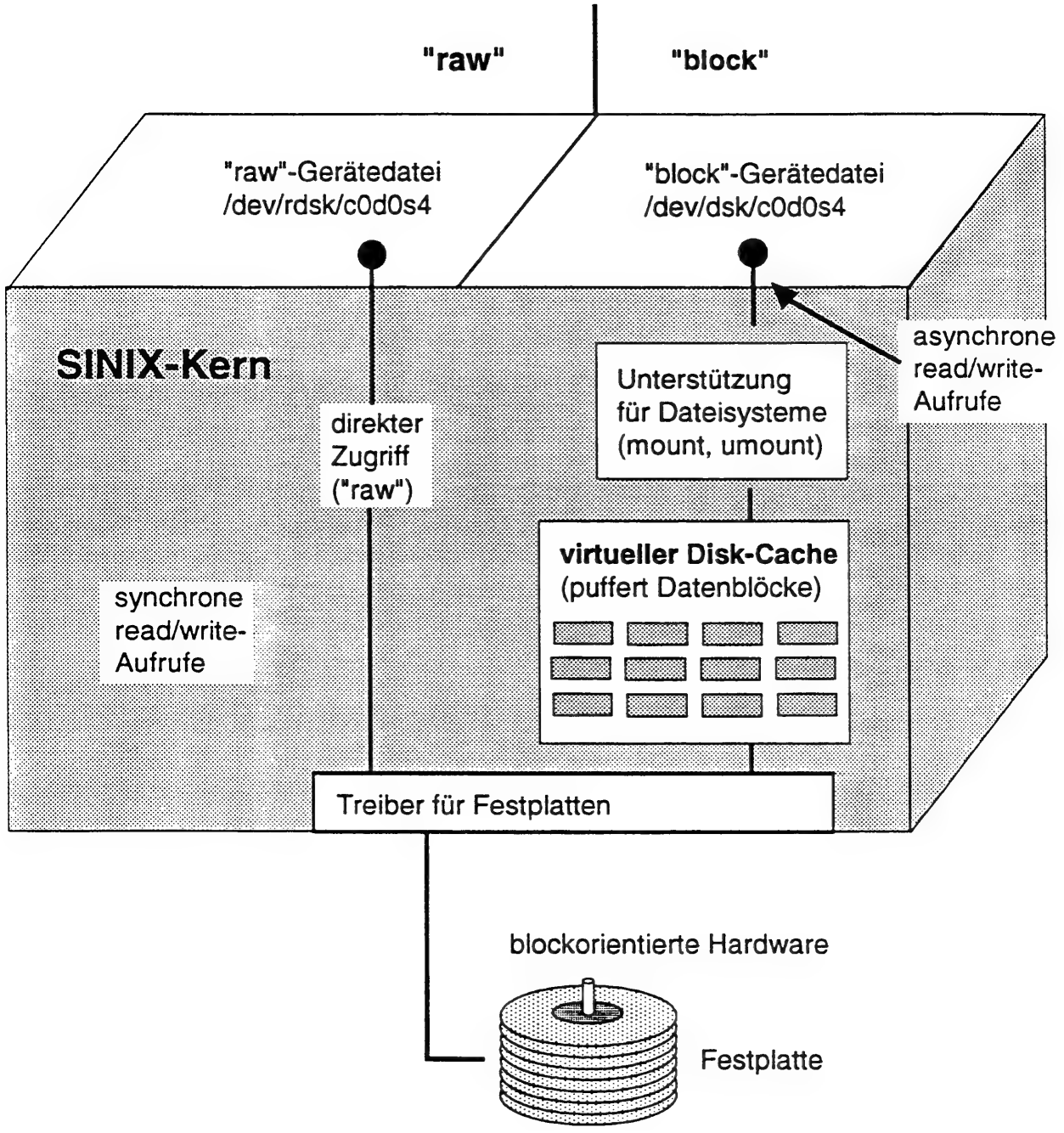
Verbrauchte Ressourcen nach der Installation:

Plattentyp	MC1355	MC1558	MC1568
/stand	1,5 MB	2,0 MB	2,0 MB
/	8,5 MB	10,0 MB	10,0 MB
/usr	36,0 MB	47,5 MB	47,0 MB
/home	0,0 MB	0,0 MB	0,0 MB
/var	1,0 MB	1,5 MB	1,5 MB
/opt	1,5 MB	1,5 MB	1,5 MB
/tmp	im / Bereich	0,0 MB	0,0 MB

Achtung:

- Der /usr-Bereich muß meist größer als 60 MB sein! Empfehlenswert sind 70 - 80 MB.
- Der /opt-Bereich sollte je nach Umfang der Anwendersoftware 40 - 80 MB groß sein.

"block" und "raw" Gerätedateien



"raw"-Gerätedateien sind **nicht** zeichenorientiert!

"block"- Gerätedateien:

Platten- und Diskettenlaufwerke sind blockorientierte Geräte, d.h. der SINIX-Systemkern kann nur ganze Datenblöcke lesen oder schreiben.

Der Systemkern puffert alle Daten, die über das Dateisystem oder über "block"-Gerätedateien gelesen oder geschrieben werden, im Arbeitsspeicher.

Die bisher besprochenen Gerätedateien für die Plattenbereiche sind "block"-Gerätedateien :

```
/dev/dsk/c0d0s1
/dev/dsk/c0d0s3
/dev/dsk/c0d0s4
```

"raw"-Gerätedateien:

Bei einigen Programmen oder Kommandos ist es sinnvoll, daß sie direkt auf die Platte zugreifen, ohne einen Zwischenpuffer zu verwenden. Diese Programme/ Kommandos verwenden die "raw"-Gerätedateien:

```
/dev/rdsk/c0d0s1
/dev/rdsk/c0d0s3
/dev/rdsk/c0d0s4
```

Label `/dev/rdsk/c0d0s0`

Vorteil : schnelle Übertragung großer Datenmengen

Anwendung : - Turbo-Informix, Informix ONLINE
- Kommandos: mkfs, fsck, ufsdump
- Labelbereich lesen

Achtung: "raw"-Gerätedateien sehen aus wie zeichenorientierte Gerätedateien, man kann jedoch nur in Blöcken zugreifen !!

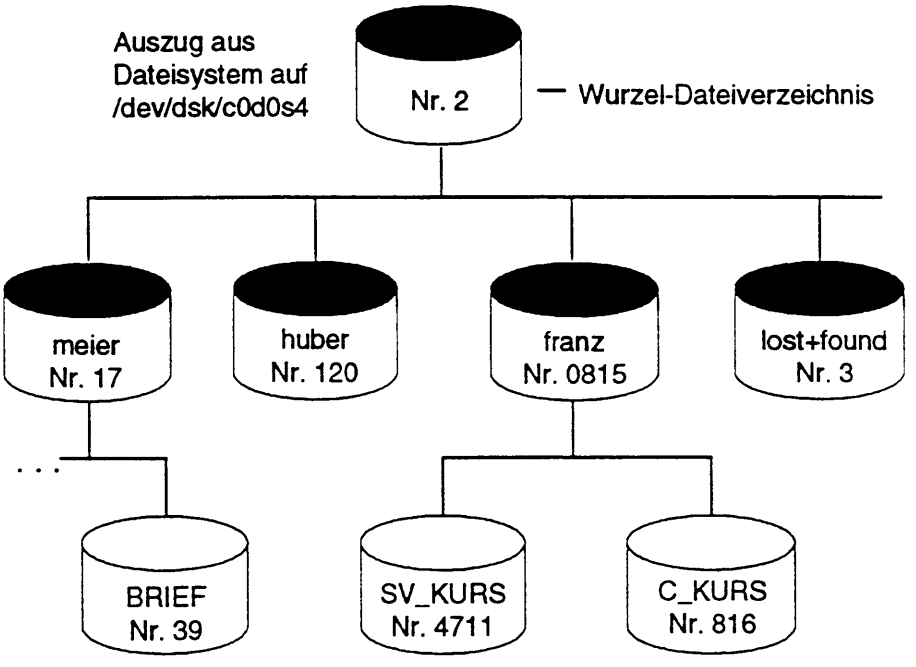
nicht verwendbar für: mount, umount

Dateibaum, Plattenbereiche, Dateisysteme

Was ist ein Dateisystem ?

a) Aus Sicht des Systemverwalters:

Ein Dateisystem ist ein **eigenständiger Dateibaum,** der ein **oberstes Dateiverzeichnis** (Indexnummer 2), Unterverzeichnisse und Dateien enthält.



Jedes Dateiverzeichnis und jede Datei besitzt eine Verwaltungsnummer (Indexnummer), die innerhalb des Dateisystems eindeutig ist.

b) Aus Sicht des Systemkerns:

Ein Dateisystem besteht aus Datenstrukturen und Datenblöcken, die der Systemkern benötigt, um Dateiverzeichnisse und Dateien zu verwalten:

- Verwaltungsinformationen ("**Superblock**")
- Inhaltsverzeichnis ("**inodes**"/"**Indexeinträge**")
- Datenblöcke für Dateien und Dateiverzeichnisse

> Ein Dateisystem benötigt ca. 15% der formatierten Kapazität eines Plattenbereichs für Verwaltungsdaten bzw. für ein Minimum an freien Blöcken und Teilblöcken (Fragmenten).

Dateibaum, Plattenbereiche, Dateisysteme

Dateisystemtypen

Bei der Installation des Betriebssystems können Sie für jeden "slice" (außer 0 und 7) festlegen:

- die Größe in Zylindern
- den Dateisystemtyp: "s5", "ufs"

  /stand hat immer den Dateisystemtyp: "bfs" !

Dateisystemtyp "bfs" (für /stand):

⇨ Primitives **Boot File System** für den Systemkern.

Dateisystemtyp "s5"

- original AT&T System V Dateisystem
- ein Superblock, eine Inode-Liste
- Datenblockgröße wahlweise: 512 B, 1 KB oder 2 KB
- Dateinamen bis 14 Zeichen
- Datenblöcke einer Datei werden im ungünstigsten Fall über das gesamte Dateisystem verteilt:

  * Schreiben ohne Optimierung (ohne CPU-Belastung)
  * langsames Lesen (Datenblöcke zusammensuchen)

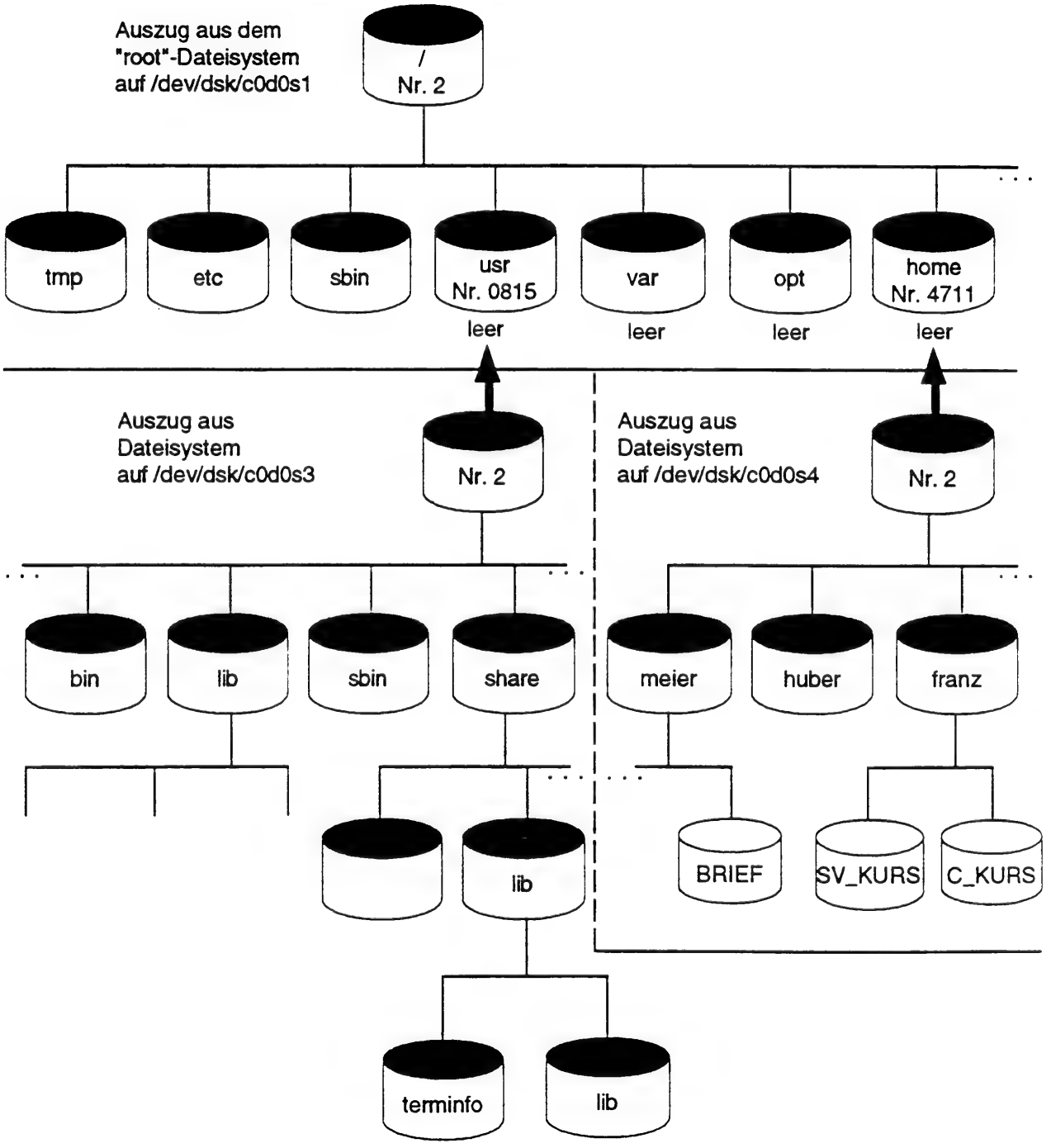
- Dateisysteme sollten regelmäßig **reorganisiert** werden (Spezialkommando: "dcopy")
- eventuell notwendig zum Ablauf älterer UNIX-Software
- **"s5"-Dateisysteme werden meist auf Disketten benutzt**

Dateisystemtyp "ufs"

- **UNIX Fast File System** vom Berkeley-UNIX
- Dateisystem ist unterteilt in Zylindergruppen
- jede Zylindergruppe besitzt eine Kopie des Superblocks, eigene Verwaltungsinformationen, eigene Inodes und eigene Datenblöcke (—> Sicherheit)
- Datenblockgröße wahlweise: 4 KB oder 8 KB
- bei kleinen Dateien wird als letzter Datenblock ein "Fragment" zwischen 1 KB und 7 KB verwendet
- Dateinamen bis 255 Zeichen
- Datenblöcke einer Datei werden möglichst in einer Zylindergruppe angelegt:
  * Schreiben kostet Rechenzeit (wegen Optimierung)
  * Daten können schnell gelesen werden
- benötigt ca. 10% freien Platz, sonst ist keine Optimierung möglich
- es ist keine Reorganisation des Dateisystems notwendig
- durch größere Blockgröße erheblich schnellerer Zugriff auf große Dateien
- **sollte immer verwendet werden, falls alle Anwendungsprogramme das "ufs"-Format für Dateiverzeichnisse unterstützen**

Dateibaum, Plattenbereiche, Dateisysteme

Ausschnitt aus den Dateibäumen der Dateisysteme:



Dateibaum, Plattenbereiche, Dateisysteme

Platteneinteilung: /etc/vfstab

In die Datei /etc/vfstab müssen Sie Plattenbereiche und Dateiverzeichnisse eintragen, die beim Hochfahren automatisch überprüft und montiert werden sollen.
Außerdem stehen Informationen zum Montieren von Installationsdisketten in dieser Datei (siehe "Installation ...").

Swap-Bereiche werden <u>nicht</u> in diese Datei eingetragen! "/dev/dsk/c0d0s2" wird automatisch als "swap"-Bereich genutzt, weitere "swap"-Bereiche müssen Sie mit dem Kommando "swap" hinzufügen.

Beispiel: **MX300-50 mit 2 MC1558 Platten:**

```
/dev/root          /dev/rroot          /        ufs   1   yes   -
/dev/dsk/c0d0s3    /dev/rdsk/c0d0s3    /usr     ufs   1   yes   -
/dev/dsk/c0d0s4    /dev/rdsk/c0d0s4    /home1   ufs   1   yes   -
/dev/dsk/c0d0s10   /dev/rdsk/c0d0s10   /stand   bfs   1   yes   -
/dev/dsk/c0d0s11   /dev/rdsk/c0d0s11   /var     ufs   1   yes   -
/dev/dsk/c0d0s12   /dev/rdsk/c0d0s12   /opt     ufs   1   yes   -
/dev/dsk/c1d0s1    /dev/rdsk/c1d0s1    /tmp     ufs   1   yes   -
/dev/dsk/c1d0s3    /dev/rdsk/c1d0s3    /home    ufs   1   yes   -
/dev/dsk/c1d0s4    /dev/rdsk/c1d0s4    /home2   ufs   1   yes   -
/proc                  -               /proc    proc  -   no    -
/dev/fd                -               /dev/fd  fdfs  -   no    -
# fuer Montieren von Installationsdisketten; Kommando: pkgadd
/dev/dsk/f0t       /dev/rdsk/f0t       /install s5    -   no    nosuid
/dev/dsk/f1t       /dev/rdsk/f1t       /install s5    -   no    nosuid
/dev/dsk/f0        /dev/rdsk/f0        /install s5    -   no    nosuid
/dev/dsk/f1        /dev/rdsk/f1        /install s5    -   no    nosuid
```

mount-Gerät ("block") | fsck-Gerät ("raw") | Montierverzeichnis | Typ des Dateisystems | fsck-pass | automatisch montieren | Optionen für mount, mit Komma getrennt

Dateisystemtyp:

s5	original UNIX System V
ufs	Berkeley "fast file system"
bfs	boot file system (nur /stand)
proc	Pseudo-Dateisystem zum Zugriff auf Prozeßdaten
fdfs	Pseudo-Dateisystem zum Zugriff auf File-Descriptoren
nfs	network file system (Berkeley)
rfs	remote file sharing (AT&T)

Kommandos für Dateisysteme

Systemverwalterkommandos finden Sie in folgenden Dateiverzeichnissen:

/sbin/
/usr/sbin/
/usr/ucb/
/etc/fs/.../

mkfs neues Dateisystem einrichten

Höchst gefährliches Kommando !
Nur auf freie Plattenbereiche anwenden !
Eventuell vorhandene Daten werden überschrieben !

```
mkfs -F ufs /dev/rdsk/c0d1s1 163200 34 15
```

mount Dateisystem montieren/einhängen

Abfragen, welche Dateisysteme wo montiert sind:

```
mount
```

Dateisystem an leeres Verzeichnis montieren:

```
mkdir DVZ
mount -F ufs /dev/dsk/c0d1s1 $HOME/DVZ
```

umount Dateisystem abmontieren

fsck -y Dateisystem überprüfen und bei Bedarf reparieren

```
fsck -F ufs -y /dev/rdsk/c0d1s1
```

Achtung: Sie dürfen "**fsck -y**" nur auf abmontierte Dateisysteme anwenden. Andernfalls wird das Dateisystem nicht repariert, sondern beschädigt !

df -k Plattenbelegung überprüfen

fsck, mount, umount

Unterschiedliche Programme für die Dateisystemtypen

/etc/fs/s5	**df**
	fsck
	fsdb
	labelit
	mkfs
	mount
/etc/fs/ufs	**df**
	fsck
	labelit
	mkfs
	mount
/etc/fs/bfs	**mkfs**
	mount
/usr/lib/fs/nfs	**mount**
	umount

Gemeinsame Kommandos unter /sbin

Sie rufen normalerweise ein Kommando unter "/sbin" auf und übergeben den Typ des Dateisystems mit dem Schalter "-F FStype" (FStype: "s5", "ufs" oder "bfs"):

```
df [-F FStype]
fsck [-F FStype] ...
labelit [-F FStype] ...
mkfs [-F FStype] ...
mount [-F FStype] ...
umount
```

Falls Sie den Dateisystemtyp nicht angeben, sucht das Kommando die Zeile in der "/etc/vfstab".
Ist kein Eintrag in der "vfstab" vorhanden, so setzt es den Typ auf "s5".
Anschließend ruft das Kommando das entsprechende Programm auf unter /etc/fs/.. oder /usr/lib/fs/.. .

mount, umount

mount -F ufs /dev/dsk/c1d0s4 /home1

Ist das Dateisystem in die Datei "/etc/vfstab" eingetragen, genügt beim "mount"-Kommando:

mount /home1

umount /home1

oder

umount /dev/dsk/c1d0s4

"umount" kann nur erfolgreich sein, wenn kein Prozeß auf das Dateisystem zugreift und wenn das Dateisystem keinen Montierpunkt eines anderen Dateisystems enthält (Reihenfolge !!!).

Falls "umount" nicht funktioniert:

```
fuser -u /dev/dsk/c1d0s4   # nachsehen, welche Prozesse
                           # auf /home1 zugreifen
```

umountall

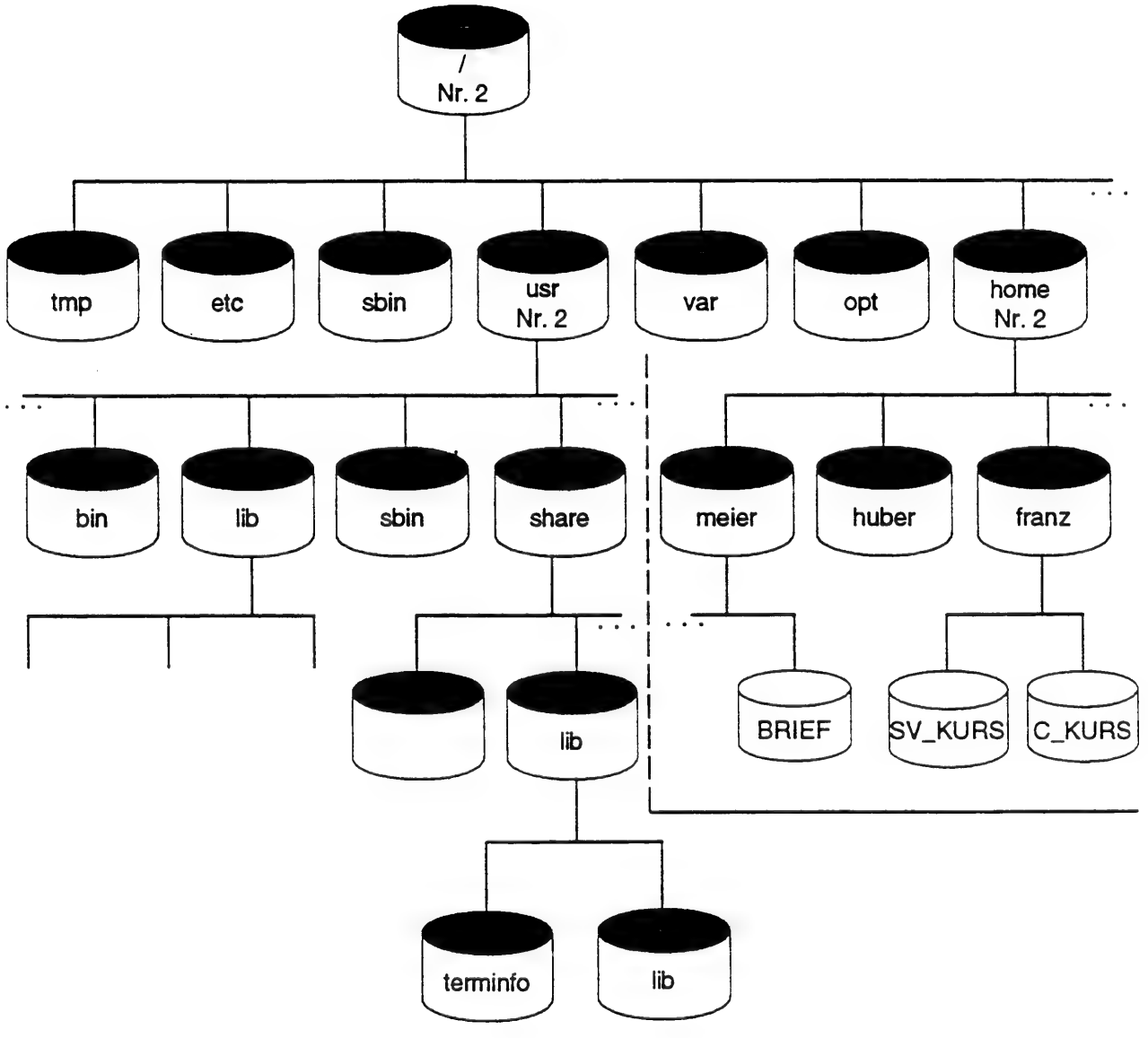
Die Prozedur **"umountall"** versucht, alle "ufs", "s5", "rfs" und "nfs" Dateisysteme abzumontieren, die in der "/etc/mnttab" eingetragen sind.
Die Dateisysteme **"/proc"**, **"/stand"** und **"/dev/fd"** werden **nicht** abmontiert !

Datei: /etc/mnttab"

Das "mount"-Kommando trägt jedes montierte Dateisystem in die Datei "/etc/mnttab" ein und "umount" löscht jedes abmontierte Dateisystem aus der "/etc/mnttab". Viele Kommandos lesen aus der "/etc/mnttab".

Dateibaum, Plattenbereiche, Dateisysteme

Dateibäume nach "mount"



Verteilte Dateisysteme (NFS)

Falls Ihr Rechner mit anderen Rechnern über TCP/IP-LAN vernetzt ist, können Sie mit Hilfe von NFS ("Network File System") Datei**verzeichnisse** eines anderen Rechners an Ihren Dateibaum "montieren".

Die Kommandos "**mount**", "**umount**" und die Datei "**/etc/vfstab**" erhalten in diesem Zusammenhang eine erweiterte Funktionalität:

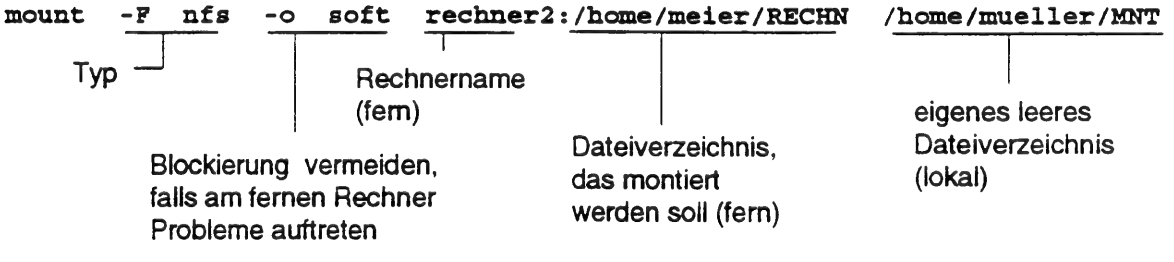
siehe: **mount(1M) (NFS)**

Datei**verzeichnisse** Ihres eigenen Rechners können Sie über das Kommando "share" für andere Rechner freigeben:

siehe Datei: **/etc/dfs/dfstab**

Datei**verzeichnis** eines anderen Rechners "montieren":

Das Dateiverzeichnis muß auf dem anderen Rechner in der Datei "**/etc/dfs/dfstab**" mit "share" für Ihren Rechner freigegeben sein.



Entsprechender Eintrag in der /etc/vfstab :

```
/dev/root /dev/rroot                    /         ufs      1    yes  -
/dev/dsk/c0d0s3 /dev/rdsk/c0d0s3        /usr      ufs      1    yes  -
/dev/dsk/c0d0s10 /dev/rdsk/c0d0s10      /stand    bfs      1    yes  -
/dev/dsk/c0d0s11 /dev/rdsk/c0d0s11      /var      ufs      1    yes  -
                                  ....
/proc                                   /proc          proc      -    no   -
/dev/fd                                 /dev/fd   fdfs           -    no   -
rechner2:/home/meier/RECHN              /home/mueller/MNT nfs  -  yes  rw,soft
```

mount-Gerät ("block") | fsck-Gerät ("raw") | Montier-verzeichnis | Typ des Dateisystems | Optionen für mount

Dateisysteme neu erzeugen ?

Normalerweise rufen Sie in SINIX V5.4 Hilfsprogramme (z.B. "disksetup") und Prozeduren auf, die für Sie das "mkfs"-Kommando starten.

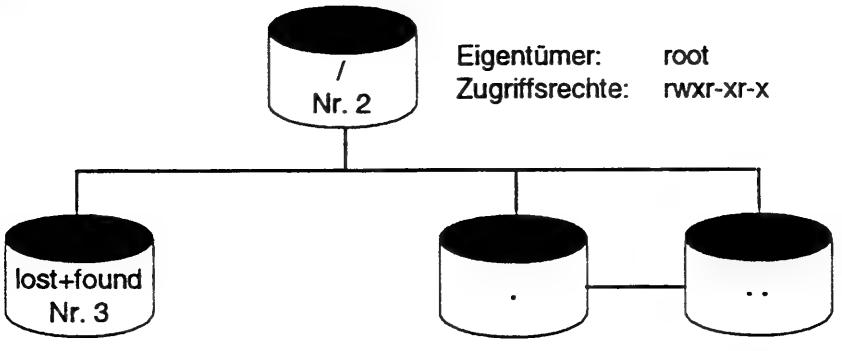
Was bewirkt "mkfs" ?

"mkfs" erzeugt im angegebenen Plattenbereich bzw. "slice" ein neues Dateisystem:

- **Eventuell vorhandene Daten werden dabei überschrieben!**
- wenn Sie "mkfs" als "root" aufrufen haben Sie noch 10 Sekunden Zeit, "mkfs" mit DEL abzubrechen

Im neuen Dateisystem richtet "**mkfs**" das oberste Dateiverzeichnis mit den Einträgen "." und ".." und ein Unterverzeichnis "**lost+found**" ein.

⇨ Mini-Dateibaum:



Beispiel für "ufs"-Dateisystem im Bereich "c1d0s4":

Vorbereitungen:

- eventuell auf dem "slice" vorhandene Daten sichern
- die Größe (length) des "slice" feststellen
- "slice" abmontieren: **umount** /dev/dsk/c1d0s4

```
mkfs -F ufs /dev/rdsk/c1d0s4 425850 34 15
#                            length nsect ntrack
```

mkfs-Kommando

Wenn Sie "mkfs" aufrufen, müssen Sie immer die Größe (length) des slice angeben (in 512-Blöcken). Zusätzlich sollten Sie auch Plattenparameter angeben, damit "mkfs" Zylindergruppen in der optimalen Größe anlegt.

"mkfs"-Parameter bei "ufs":

nsect	Anzahl Sektoren pro Spur		
ntrack	Anzahl Köpfe bzw. Spuren pro Zylinder		
bsize	Dateisystemblockgröße:	8192 oder 4096	
fragsize	Fragmentgröße:	1024	
cgsize	Zylinder pro Zylindergruppe:	Standard	16 (normalerweise 2*bsize)
free	min. freier Platz in Prozent:	Standard	10
rps	Umdrehungen der Platte/Sek:	Standard	60
nbpi	Anzahl Datenbytes/Inode:	Standard	2048

Parameterangabe für "mkfs" über den Schalter "-o"

```
mkfs -F ufs -o nsect=34,ntrack=15 /dev/rdsk/c1d0s4 425850
                                                   length
```

Parameterangabe für "mkfs" über eine Liste

```
mkfs -F ufs  /dev/rdsk/c1d0s4 45850    34    15    4096  1024   8 ...
#                             length nsec  ntrack bsize fragsize cgsize ...
#                    unbedingt die Reihenfolge der Werte beachten!
```

Die Größe **"length"** müssen Sie immer angeben!

Parameter eines existierenden Dateisystems ausgeben:

```
mkfs  -F  ufs  -m  /dev/rdsk/c0d0s13
```

Anzahl der "inodes"

Die Anzahl der "inodes", die "mkfs" anlegt, begrenzt die maximale Anzahl von Dateien und Verzeichnissen, die Sie später anlegen können

Anzahl "inodes" überprüfen:

```
df -F ufs -o i /dev/rdsk/c1d0s4
```

Wie viele "inodes" legt "mkfs" an?

Bei "ufs"-Dateisystemen gibt es 2 Regeln:

1) **Parameter: nbpi** Anzahl Datenbytes, für die "mkfs" eine "inode" einrichtet

   Standardformel: ANZAHL = length *512 / nbpi
   Standardwerte: nbpi = 2048

2) **Pro Zylindergruppe max. 2048 "inodes"**

   Sind 2048 "inodes" pro Zylindergruppe erreicht, wirkt der Parameter "nbpi" nicht mehr.

"mkfs"-Parameter bei "s5"

```
mkfs -F s5 [-b bsize] /dev/rdsk/... length [:inodes] [gap blocks/cyl.]
```

"length" (in Sektoren von 512 Bytes) müssen Sie angeben.
"bsize" (Dateisystemblockgröße: Standard 1024), die Anzahl der Inodes und Werte für "gap" und "blocks/cylinder" können Sie angeben:

Dateisystem auf Diskette:

```
mkfs -F s5 /dev/rdsk/f03ht 2844:711 2 36 # Diskette
```

Parameter eines existierenden Dateisystems ausgeben:

```
mkfs -F s5 -m /dev/rdsk/f03ht
```

Dateibaum, Plattenbereiche, Dateisysteme

Aufgabe 3

Der Kursleiter teilt Ihrer Arbeitsgruppe einen freien Plattenbereich zu, den Sie für die Aufgaben verwenden können.

Im Dateiverzeichnis "/home/sv0/sbin" stehen für Sie spezielle Kommandos für "mount", "umount" und "chown" zur Verfügung, die Ihnen gezielt "Super-User"-Rechte verleihen.

a) Erstellen Sie sich folgende ".profile"-Datei:

```
PATH=/home/sv0/sbin:$PATH:/sbin:/usr/sbin:
LANG=De_DE.646
export PATH LANG
```

Melden Sie sich anschließend neu an. Rufen Sie kein Kommando mit dem absoluten Pfadnamen auf (nur "mount", nicht "/sbin/mount").

b) Stellen Sie mit "prtvtoc" und "df" die Platteneinteilung für die Systemplatte(n) des Kursrechners fest.
Wie groß sind die einzelnen "slices"?
Wie heißen die Gerätedateien ?
Ist die Einteilung sinnvoll?

1)

	7	1	2	3	4	10	11	12	13

2)

	7								

c) Erzeugen Sie mit "mkfs" in Ihrem Plattenbereich ein neues leeres "ufs"-Dateisystem und prüfen Sie es mit "fsck".

d) Erzeugen Sie ein Datei**verzeichnis** in Ihrem Login-Verzeichnis und montieren Sie Ihr Datei**system** mit "mount" an dieses Datei**verzeichnis.**
Überprüfen Sie Indexnummer, Zugriffsberechtigungen und Eigentümer Ihres Verzeichnisses vor und nach dem "mount"-Kommando ("ls -ild DVZ").

e) Ändern Sie den Eigentümer mit dem speziellen "chown".

f) Wechseln Sie in Ihr Dateiverzeichnis und kopieren Sie die /etc/passwd und Ihre "$HOME/.profile"-Datei in Ihr Dateisystem.
Prüfen Sie mit "ls -ial" die Indexnummern der neu angelegten Dateien und Dateiverzeichnisse.

g) Montieren Sie das Datei**system** mit "umount" wieder ab und überprüfen Sie, ob das Dateiverzeichnis leer ist. Unter welchen Umständen kann der "umount" ein Dateisystem nicht abmontieren ?

h) Überprüfen Sie Ihr Dateisystem mit "fsck".

i) Montieren Sie Ihr Dateisystem an ein **anderes** Dateiverzeichnis in Ihrem HOME-Verzeichnis und prüfen Sie den Inhalt.

j) Diese Aufgabe funktioniert nur, wenn Sie nicht mit der "ksh" arbeiten.
Montieren Sie das Dateisystem mit "umount" wieder ab und geben Sie folgende Kommandos ein:

```
cd $HOME
echo "Dies ist mein HOME-Verzeichnis" > datei1
ls -il datei1
cd ..
mount -F ufs /dev/????  $HOME
                 |
                 └──── Ihre Gerätedatei
cd $HOME
ls -ial
```

Erklären Sie das Ergebnis! Wo ist die "datei1" ?

Was passiert, wenn Sie sich ab- und wieder anmelden ?
Was müssen Sie tun, um Ihre "datei1" wieder zu erhalten ?

Zusatzaufgabe: symbolische Verweise

k) Überlegen Sie sich alle Unterschiede zwischen normalen Verweisen (**"Links"**, **"ln"**-Kommando) und symbolischen Verweisen (**"ln -s"**). Probieren Sie dazu folgende Kommando-Folgen aus und erklären Sie die Ergebnisse:

Kommadofolge für normalen Verweis:

```
echo Hallo > dat1          # Datei erzeugen
ln dat1 dat2               # Verweis auf die gleiche Datei
ls -il dat?                # Indexnummern und Verweiszähler
                           # überprüfen
echo Aenderung >> dat1     # dat1 ändern
cat dat2                   # dat2 ansehen

rm dat1                    # einen Verweis löschen
cat dat2                   # Datei ist noch da

ln dat2 /tmp               # warum funktioniert das nicht ??
```

Kommandofolge für symbolischen Verweis:

```
echo Hallo2 > orig          # Datei "orig" erzeugen
ln -s orig verw             # symbolischer Link "verw"
ls -il orig verw            # Indexnummern und Verweis-
                            # zähler überprüfen

echo Aenderung >> orig      # Datei ändern
cat verw                    # Verweis ansehen
DVZ=`pwd`
ln -s $DVZ/orig /tmp/verw   # anderes Dateisystem

rm orig                     # Originaldatei löschen
ls verw                     # ist die Datei noch da ?
cat verw                    # eigenartige Fehlermeldung

cd /etc                        # A c h t u n g
ln -s passwd $HOME/testdat     # f a l s c h ! ! !
cd $HOME
cat testdat                 # wo liegt der Fehler ?
```

Zusätzliche Platte konfigurieren

Das Betriebssystem wird bei MX300-Rechnern bereits im Werk **auf der 1. Platte** vorinstalliert.
Um eine zusätzliche Platte zu aktivieren, müssen Sie:

- die UNIX Partition einrichten ("/sbin/fdisk")
- die Platteneinteilung und Dateisystemtypen festlegen ("/sbin/disksetup -I")

> **MX500-90:**
>
> Haben Sie in einen MX500-90 eine neue Platte eingebaut, so müssen Sie das Kommando **"diskadd"** aufrufen, um die Gerätedateien anzulegen. "diskadd" ruft dann automatisch "fdisk" und "disksetup" auf.

Beispiel für 2. Platte am 2. Controller:

```
fdisk   /dev/rdsk/c1d0s0
```

Sollte die Liste der aktiven Partitionen nicht leer sein, so sollten Sie zunächst die aktive Partition **löschen** (Auswahl 3).
Anschließend wählen Sie die Auswahl 1 und erzeugen **eine** neue UNIX Partition, die 100% der Platte belegt. Die UNIX-Partition muß aktiv sein.

```
disksetup -I    /dev/rdsk/c1d0s0
```

"disksetup" stellt Ihnen verschiedene Fragen:

- Anzahl der von Ihnen gewünschten "slices"
- absolute Pfadnamen der Dateisysteme (z.B.: /home1)
- gewünschter Dateisystemtyp: s5, ufs, na (no access)
  "na" ist z.B. sinnvoll für einen 2. "swap"-Bereich oder bei "raw"-Bereichen für Datenbanken
- **Größe der "slices" in Zylindern oder in Sektoren (je nach SINIX-Version)**

Eine genauere Beschreibung finden Sie in der SINIX-Freigabemitteilung.

disksetup

Platteneinteilung und Dateisystemtypen festlegen:
(vgl. vorhergehende Seite)

Sie können dem "**disksetup**"-Programm eine Datei angeben, in der Sie Standardwerte für die Platteneinteilung hinterlegt haben:

```
disksetup -d stdwert.dat -I /dev/rdsk/c1d0s0
```

Aufbau einer Datei mit Standardwerten für "disksetup":

```
 1    /roots    ufs    15M    #   ca. 15 MB
 2    swap2     -      2m     #   ca. 2 x Hauptspeicher
 3    /var_n    ufs    40M    #   ca. 40 MB
 4    /home1    ufs    60W    #   ca. 60 % vom Rest
 5    /home2    ufs    40W    #   ca. 40 % vom Rest
 6    /tmp_n    ufs    10M    #   ca. 10 MB
10    /stands   bfs    5M     #   ca. 5 MB
```

Dateisystemtyp oder "-"

slice — mount-DVZ

In einer Standardwertdatei für disksetup sollte mindestens eine Zeile mit der Größenangabe "W" (Prozent vom Rest) vorkommen.

Was macht "disksetup" ?

- Platteneinteilung auf die Platte schreiben (VTOC)
- Dateisysteme mit "**mkfs**" einrichten
- mount-Dateiverzeichnisse anlegen
- "**/etc/vfstab**" erweitern
- Dateisysteme montieren

"disksetup" aktualisiert jedoch nicht die Datei "/etc/partitions" !

Mit "disksetup" können Sie auch das "boot"-Programm in den Zylinder 0 der Platte schreiben:

```
disksetup -b /etc/boot /dev/rdsk/c0d0s0
```

Platteneinteilung nachträglich ändern

In SINIX V5.4/UNIX V.4 können Sie die Platteneinteilung auch nachträglich ändern, z.B.:

- einen slice in mehrere slices aufteilen
- zwei slices zu einem slice zusammenfassen

Solche Änderungen erfordern große Vorsicht, da Sie den Startsektor und die Länge (size, Anzahl Sektoren) der slices selbst berechnen müssen. Die Größe eines "slices" sollte immer ein Vielfaches der Zylindergröße sein !!!

Sie haben 2 Kommandos zur Verfügung:

mkpart mit Schalter "-p" und "-P"(von mkpart wird jedoch abgeraten)
"mkpart" verwendet die Datei "/etc/partitions"

edvtoc -f
"edvtoc" verwendet eine Datei im Format, das Sie mit "prtvtoc -f .... " erzeugen können.

Beispiel für "edvtoc":

Die slices 4 und 5 (/home1 und /home2) aus dem letzten Beispiel zu einem slice zusammenfassen:

1. Schritt: Daten sichern, /home1 und /home2 abmontieren

2. Schritt:
```
prtvtoc -f vtoc1.dat /dev/rdsk/c1d0s0
```

3. Schritt: Sie können nun die Datei "vtoc1.dat" editieren und die neue Platteneinteilung eintragen. Das Format der Datei "vtoc1.dat" finden Sie auf der nächsten Seite.

4. Schritt: neue Platteneinteilung in VTOC schreiben:

```
edvtoc -f vtoc1.dat /dev/rdsk/c1d0s0
```

5. Schritt: neues Dateisystem mit "mkfs" erzeugen, "/etc/vfstab" modifizieren

Dateiformat: "edvtoc -f" und "prtvtoc -f"

Ausgabedatei von "prtvtoc -f vtoc1.dat /dev/...."
vtoc1.dat:

#SLICE	TAG	FLAGS	START	SIZE
0	0x5	0x201	510	621180
1	0x4	0x200	544	31110
2	0x4	0x201	31654	49470
3	0x4	0x200	81124	82110
4	0xb	0x200	163234	255510
5	0x4	0x200	418744	170340
6	0x4	0x200	589084	20910
7	0x1	0x201	510	34
8	0x0	0x0	0	0
9	0x0	0x0	0	0
10	0x4	0x200	609994	11696
11	0x0	0x0	0	0

Die Bedeutung der Spalten “TAG” und “FLAGS” finden Sie in der Datei “/usr/include/sys/vtoc.h”:

TAG:

0x01	Boot slice
0x02	Root filesystem
0x03	Swap filesystem
0x04	Usr filesystem
0x05	full disk
0x09	Stand slice
0x0a	Var slice
0x0b	Home slice
0x0c	dump slice

FLAGS:

0x01	Unmountable partition
0x10	Read only
0x100	Partition open (for driver use)
0x200	Partition is valid to use

Bei der Größe "SIZE" sollten Sie immer ein Vielfaches eines Zylinders angeben.

Wie müssen Sie die Datei “vtoc1.dat” verändern, um die slices 4 und 5 in einem slice zusammenzufassen ?

"swap"-Bereiche konfigurieren

"**slice**" 2 der 1. Platte ist immer "**swap**"-Bereich !

Haben Sie eine 2. Platte nachträglich installiert, ist es sinnvoll, einen 2. "**swap**"-Bereich auf der 2. Platte einzurichten.

Mit dem Kommando **"swap"** können Sie einen zusätzlichen "swap"-Bereich aktivieren, deaktivieren oder die Liste der aktiven "swap"-Bereiche anzeigen lassen:

```
swap -a /dev/dsk/..... 0 length # aktivieren
```

- Sie müssen die **"block"**-Gerätedatei und die "length" (in 512 Byte-Blöcken) eines freien "slices" oder eine normale Datei angeben. Falls Sie eine normale Datei angeben, muß sie bereits die Größe von "length"*512 besitzen.
- der Parameter 0 steht für den Startblock, ab dem der "slice" zum "paging" und "swapping" verwendet wird
- das Kommando muß bei jedem Systemstart neu gestartet werden

```
swap -d /dev/dsk/..... 0      # "swap" deaktivieren

swap -l                       # Liste ausgeben
```

Wie gehen Sie vor ?

- Wählen Sie einen freien "slice", der nicht für ein Dateisystem genutzt wird.

  Beispiel: **/dev/dsk/c1d0s2** (2. Platte am 2. Controller)

- Beim MX300 schreiben Sie eine rc-Prozedur "addswap", die das Kommando **"swap"** aufruft (—>/etc/init.d/). Wie Sie dies tun können, erfahren Sie im Kapitel "rc-Dateien".
- Beim **MX500-90** tragen Sie zusätzliche swap-Bereiche in die Datei "**/etc/swaptab**" ein. Eine entsprechende rc-Prozedur existiert bereits.

Sinnvolle Plattenaufteilung

Wenn Sie nur eine Platte besitzen, ist die Plattenaufteilung im wesentlichen vorgegeben.

Haben Sie jedoch 2 Platten (möglichst jeweils mit eigener Plattensteuerung), dann hängt die Leistungsfähigkeit Ihres Rechners davon ab, ob Sie die Plattenzugriffe gleichmäßig auf beide Platten verteilen.

Sie sollten außerdem dafür sorgen, daß alle Dateisysteme maximal bis 90 % belegt sind, da Sie sonst Geschwindigkeitseinbußen in Kauf nehmen müssen.

Beispiel:

1. Platte:

	s7	c0d0s10 /stand	c0d0s2 swap	c0d0s1 /	c0d0s3 /usr	c0d0s4 /home1	c0d0s11 /var	c0d0s12 /opt

2. Platte mit eigener Plattensteuerung (Controller):

	s7	c1d0s1 /tmp	c1d0s2 swap	c1d0s3 /home	c1d0s4 /home2

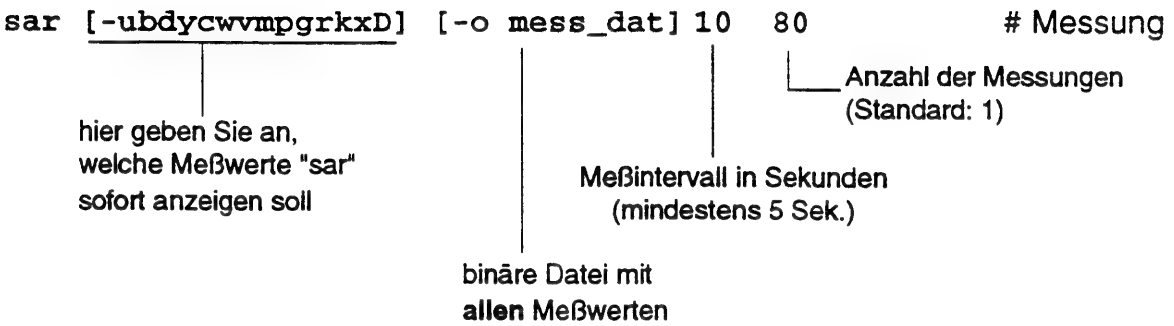
reserviert

Es wäre besser, wenn auch "/var" auf der 2. Platte liegen würde !

/tmp wurde hier bei der Installation nicht angelegt, aber später auf der 2.Platte eingerichtet!

Systemaktivitäten messen:

sar: system activity reporter

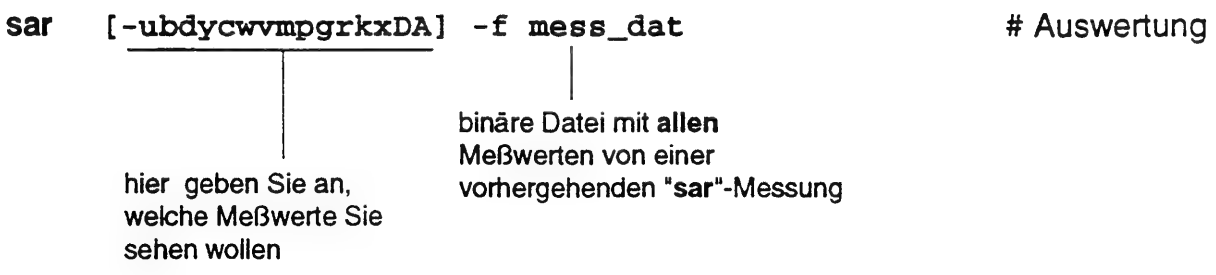


Der UNIX-Kern enthält zahlreiche Zähler, die bei den verschiedensten Systemaktivitäten hochgezählt werden.
"**sar**" liest alle Zähler und speichert sie in der binären Ausgabedatei, die Sie über den Schalter "**-o**" angegeben haben.

Über Schalter können Sie festlegen, welche Systemaktiviäten "**sar**" sofort auf der Standard-Ausgabe ausgibt. Sie sollten zunächst maximal einen Schalter angeben, weil sonst die Ausgabe nicht lesbar ist.

Auswertung der Messdatei mit "sar":

Sie haben mit dem ersten "sar"-Kommando eine binäre Messdatei erzeugt, die Sie nun nachträglich auswerten können:



Bei der Auswertung einer Meßdatei können Sie mehrere Schalter gleichzeitig angeben. Sie erhalten dann mehrere Auswertungstabellen hintereinander.

Bedeutung einiger Ausgabe-Schalter:

-u	CPU-Benutzung (Standard, falls kein Schalter angegeben ist)
-D	D wie "**-u**" aber mit getrennter Berücksichtigung von fernen (remote) Rechnern im Netz
-d	Plattenaktivitäten
-y	Terminal Ein-/Ausgabe
-c	Systemaufrufe
-cD	wie "**-c**" aber mit getrennter Berücksichtigung von fernen (remote)-Rechnern im Netz
-w	"**swapping**" und Prozeßwechsel
-p	"**paging**"-Aktivitäten
-r	freie 4 KB Seiten im Arbeitsspeicher und freie 4 KB Blöcke im "**swap**"-Bereich
-v	Auslastung von Systemtabellen (Prozeß-Tabelle, Inode-Tabelle, offene Dateien, Locking-Tabelle)

Die Größenangabe SZ bei "**ps -el**" erfolgt ebenfalls in 4 KB Seiten !

Beispiele für sar

Messung:

```
sar -o mess_dat 10 20          # 20 Messungen mit 10 Sekunden-
                               # Intervall durchführen
```

Sie sollten das Messintervall normalerweise größer als 5 Sekunden wählen.

Während der Messung gibt "sar" standardmäßig die CPU-Benutzung aus:

10:02:17	%usr	%sys	%wio	%idle
10:02:22	3	34	11	52
10:02:27	2	20	12	66
10:02:32	1	29	70	0
10:02:37	4	30	49	17
	...			
10:03:07	4	25	15	56
Average	2	24	22	51 Mittelwert

% für CPU-Zeit im **"user-mode"**

% CPU-Zeit im **"system-mode"**

warten auf **"block"**-E/A

CPU war untätig

Auswertung der Messdaten:

```
sar -r -f mess_dat             #Beispiel: freier Speicher & "swap"
```

10:02:17	freemem	freeswp
10:02:22	410	7691
10:02:27	472	7757
10:02:32	280	7756
10:02:37	365	7715
...		
10:03:07	476	7732

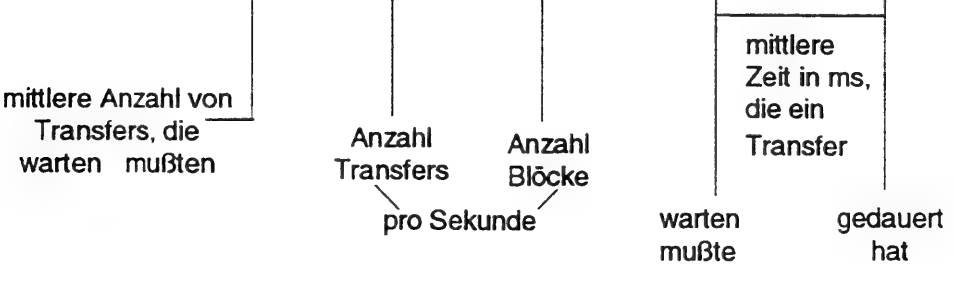
jeweils in 4 KB Seiten !

Auswertung der Messdateien:

```
sar -d -f mess_dat   #  Beispiel:   Platten-Aktivität
```

Beispiel für MX500:

```
SINIX-M mxi500rzs 5.40 A009 MX500I    02/12/92

09:05:22   device   %busy    avque    r+w/s    blks/s    avwait    avserv    devname

09:05:32   dsk-0    14       1.5      4        51        17.4      37.1      ssd(0,0)
           dsk-2    1        1.0      0        4         0.0       23.3      ssd(1,0)

09:05:42   dsk-0    15       1.9      5        55        28.2      29.8      ssd(0,0)
           dsk-2    11       1.0      5        73        0,4       21.6      ssd(1,0)

09:05:52   dsk-0    35       3.5      8        115       113.9     46.2      ssd(0,0)
           dsk-2    14       1.0      7        90        0.1       20.0      ssd(1,0)

09:06:02   dsk-0    7        1.9      2        22        35.6      38.9      ssd(0,0)
           dsk-2    2        1.0      1        8         0.0       35.0      ssd(1,0)

09:06:12   dsk-0    28       3.9      7        80        113.7     39.2      ssd(0,0)
           dsk-2    16       5.1      4        52        178.6     43.2      ssd(1,0)
                             ...


09:08:22   dsk-0    20       2.9      5        69        77.4      40.0      ssd(0,0)
           dsk-2    9        1.0      4        47        0.0       20.5      ssd(1,0)

09:08:32   dsk-0    18       2.9      4        63        88.5      46.2      ssd(0,0)
           dsk-2    1        1.0      0        3         0.0       30.0      ssd(1,0)

09:08:42   dsk-0    18       1.8      4        48        32.0      38.9      ssd(0,0)
           dsk-2    23       2.5      5        75        69.0      46.2      ssd(1,0)

Average    dsk-0    24       1.9      8        62        26.3      29.9      ssd(0,0)
           dsk-1    0        1.0      0        0         0.3       14,7      ssd(0,1)
           dsk-2    15       1.4      5        49        11.0      29.0      ssd(1,0)
```



truss: Systemaufrufe protokollieren

Aufruf:

truss -o protdat KOMMANDO # Kommando starten und
protokollieren

oder

truss [-o protdat] -p PID # laufenden Prozeß mit
NR.: PID protokollieren

Beispiele:

a) ***truss -o protdat ced testdat***
pg protdat # Protokolldatei ansehen

b) ***truss -p 1*** *# init-Prozeß protokollieren*

Das "truss" Kommando ist beschrieben im Handbuch "Kommandos V5.4, Band 2, L-Z".

Die Auswertung der Protokolldateien erfordert allerdings sehr gute Kentnisse der C-Systemaufrufe und führt daher über das Thema dieses Kurses hinaus.

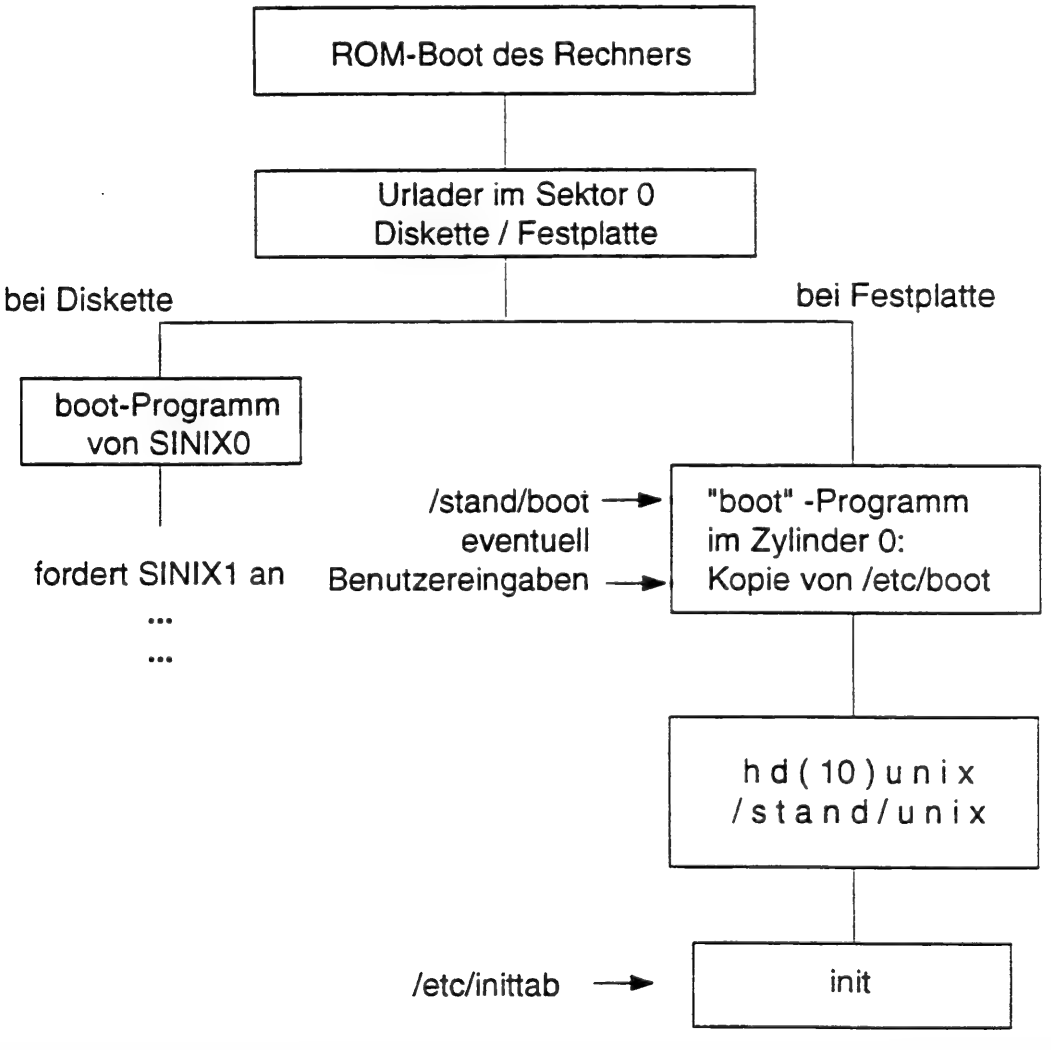
Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

Boot von Diskette oder Platte

Installationsdisketten (MX300):

SINIX0: Diskette mit "boot"-Programm
SINIX1: Miniroot-Dateisystem ("s5") mit Systemkern
SINIX2: Dateisystem ("s5") mit Kommandos und Installationsprozeduren

Systemstart beim MX300:



Systemstart beim MX500:

Beim MX500 bestimmen Sie über den Hardwaremonitor, ob der Rechner vom Band oder von der Platte bootet (siehe Kapitel 6 des Kursordners)

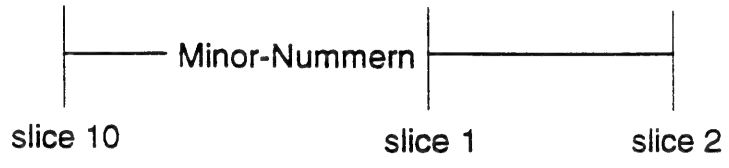
Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

boot-Parameter

MX300: Datei /stand/boot (nicht /etc/default/boot)

`AUTOBOOT=YES` bei "YES" wartet "boot" nicht auf Benutzereingaben
`NO` sondern lädt sofort das Programm, das in DEFBOOTSTR angegeben ist:

```
DEFBOOTSTR=hd (10,0) unix root=hd(1)   swap=hd(2)
```



`TIMEOUT=15` falls `AUTOBOOT=NO`, wartet "boot" 15 Sekunden auf eine Benutzereingabe und lädt dann das Programm aus `DEFBOOTSTR`.

`rootfstype=ufs` Typ des "root"-Dateisystems ("s5", "ufs")

Benutzereingaben für boot:

- im Fehlerfall: falls das Programm in `DEFBOOTSTR` nicht geladen werden kann

- bei `AUTOBOOT=NO`

Festplatte:	—>	hd(m,o) [pfadname]	
	oder	hd(m) [pfadname]	m-Minor-Nummer
Diskette:	—>	fd(m,o) [pfadname]	o-Offset (meist 0)
	oder	fd(m)[pfadname]	

Beispiel: "booten" von Diskette mit "Ram-Disk":

```
fd(7)unix root=ramd(0) swap=ramd(1)
```

MX500-90 : siehe bootflags des Hardwaremonitors ("/stand/boot" spielt keine Rolle)

Typischer Bootstring für MX500-90:

```
sp(0,10)unix root=ssd(m1) swap=ssd(m2) rootfstype=ufs
```

m1, m2 == Minornummern der slices

Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

Meldungen beim Systemstart

> Alle Meldungen des SINIX-V5.4-Kerns finden Sie bei einem laufenden SINIX-System am Ende der Datei: **"/var/adm/klog.msg"**

- Meldungen beim Systemstart
- Fehlermeldungen im laufenden Betrieb

Beispiel für MX 300:

```
Wed Feb 12 07:36:21 MET 1992
total real memory        = 16777216
total available memory   = 14553088

SINIX-L Release 5.40  Version A20N5

    ....

  device     address    iobase    ipl    maps[at]   comment

  :Proc      -          -         -      -          i486: typ 00, frq 00, gs 01
  :Mbus      FC000000   FD000000  -      FFF[000]
  +exos0:    HW 0.0 NX 5.5, ether 08-00-14-81-30-31
  :exos0     00001A00   00000000  01(5)  06F[000]
  +hd0       -          -         -      -          drive 0: MC1664
  +fd0       -          -         -      -          drive 2: Floppy
  +fd1       -          -         -      -          drive 3: Floppy
  +hd1       -          -         -      -          drive 0: MC1664
  :hd0       000073F8   00000000  -      100[0BF]   Storager FW 45
: :hd1       000075F8   00000001  -      100[1BF]   Storager FW 45
  +it0       -          -         -      -          drive 0: TDC 3650
  :is_ts0    000073FC   00000000  -      100[0BF]   Storager FW 45
  :is_ts1    000075FC   00000001  -      100[1BF]   Storager FW 45
  :ncr0      00004800   00000000  04(5)  082[323]   FW D Id 7 Tgs -
  :s_0: FW 2, memsize 0x000F0000
  :s_0       00001900   00ED0F20  05(5)  -
  :sc0       E0000100   -         -      -
  :sr0       00D80000   00001400  -      -          fw 0x82, size 16 KB
  :sr1       00D84000   00011500  -      -          fw 0x03, size 16 KB
  #acc       -          -         -      -          unit(s) 2
  #exa       -          -         -      -          unit(s) 3
  #fd        -          -         -      -          unit(s) 0
  #ld12      -          -         -      -          unit(s) 0
  #shd       -          -         -      -          unit(s) 6
  #sv        -          -         -      -          unit(s) 15
  #sx        -          -         -      -          unit(s) 5
  #vp        -          -         -      -          unit(s) 0
  Collage Line Discipline and Vtty Driver                linecnt = 4
```

Arbeitsspeicher-ausbau
Version des Systemkerns

Ethernet Baugruppe

1. Platte
Disketten

2. Platte am 2.Controller
2 Contoller

Streamer
(die gleichen Controller)

NCR-Hostadapter

DFÜ-Prozessor

E/A-Prozessoren

Treiber für "Juke-Box"
Treiber für Exabyte

Treiber für externe Platten

Treiber für Spiegelplatten

Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

klog.msg für MX500

```
0:total real memory      =   33554432
0:total available memory =   29995008
0:
0:3 MULTIBUS Adapters; slic(dma-range) 48(128M) 49(128) 50(128).
0:Not  using MULTIBUS Adapters: slic 50.
SCED 0: SLIC 22 Version 43 (console server)

Devices on SCED boards
UNIT    SCED    CHANNEL    SCSI(TARGET/LUN)    INTR(BIN/vec)
co 0    0       3          -/-                 2/ 0
                ...
ct 0    0       64         4 / 0               5 / 0
se 0    0       1          -/-                 4 / 0
se 1    0       2          - /-                4 / 1
sm 0    0       8          - / -               4 / 2
                ...

MBAd / pseudo devices:
device    MBAd addr  iobase   ipl      maps[at]  comment
:adp32.0  0 000000   006000   4(5,9)   24[00]    FW Rev G Id 7 Tgs 0
+fd0      -  -       -        -        -         drive 2: Floppy
:hd0      0 000000   0031F8   2(5,A)   2D[26]    Storager FW 45
:si0      0 005000   080000   5(4,3)   12[53]    fw 0x8c, size 512
:si1      0 015100   084000   5(4,3)   12[65]    fw 0x8c, size 512
:sr2      0 024200   088000   -        -         SX, fw 0x47, sz 16 KB
:sr14     0 0E4E00   0B8000   -        -         fw 0x82, size 16 KB
0: xmt0: Not alive / binary configured. Ignored
:MBAd0    0          -        -        89[77]    slic 48, 256 maps a 128M range
:MBAd1    1          -        -        00[00]    slic 49, 256 maps a 128M range
#acc-     -          -        -        -         max. number of units: 3
+exa1t0   -          -        -        -         EXABYTE EXB-8200   FW 425A
#exa      -          -        -        -         max. number of units: 3
#fd       -          -        -        -
#fl       -          -        -        -
#lad      -          -        -        -         max. number of units: 28
#shd      -          -        -        -         max. number of units: 24
#sv       -          -        -        -         max. number of units: 15
#vp       -          -        -        -

Disk devices on SPA: ssd driver
UNIT    SPA-CT    CHANNEL    TARGET    INTR(BIN/VEC)    comment
0       0         0          0         6 / 0            MICROP  1588-15MB1036511
1       0         0          1         6 / 1            MICROP  1588-15MB1036511
2       0         1          0         6 / 2            MICROP  1588-15MB1036511
```

Arbeitspeicher-ausbau

Geräte an SCED-Baugruppe:

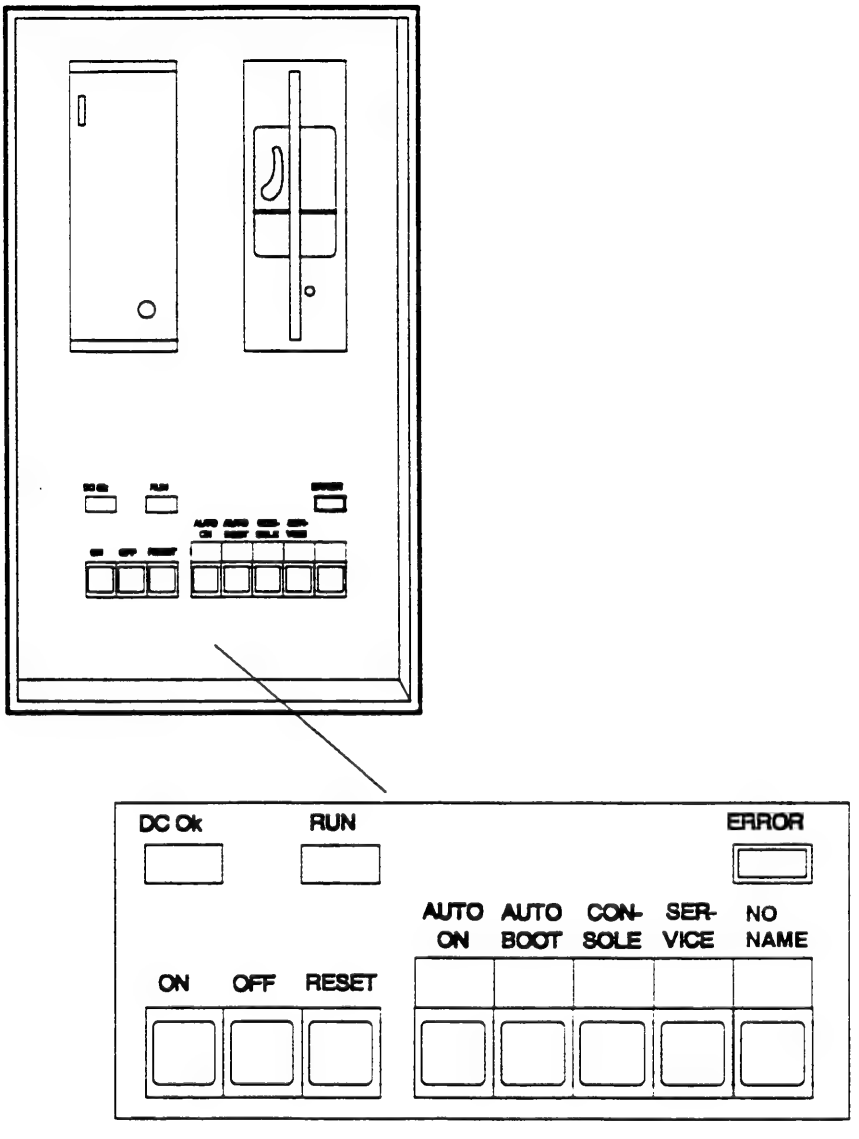
- Konsole
- Streamer
- Ethernet-Anschluß
- batteriegepufferter Speicher

Geräte an den Multibusadaptern

- SCSI-Hostadapter
- Diskette am Storager
- SIM-Baugruppen
- E/A-Prozessor
- Multibusadapter
- Treiber für Juke-Box
- Exabyte

synchrone SCSI-Platten:

Bedienfeld MX500



AUTO-ON
Anlage wird nach Stromausfall automatisch wieder eingeschaltet

AUTO-BOOT
Nach dem Einschalten oder RESET wird automatisch gebootet.
Der MX500 bleibt nicht im Monitor-Modus.

CONSOLE
Konsolterminal am Steckplatz C

SERVICE
Konsolterminal am Steckplatz S

Verhalten im Fehlerfall/Notfall beim MX300:

Falls Sie sich an keinen Terminal mehr anmelden können und "nichts mehr läuft", gehen Sie wie folgt vor:

- drücken Sie die "?"-Taste (Diagnose-Monitor)
- warten Sie, bis der Rechner jegliche Aktivität eingestellt hat
- geben Sie an der Tastatur <RETURN> ein und warten Sie bis die Meldung erscheint:

  syncing disk ... done

- jetzt drücken Sie RESET oder OFF

Wenn das Betriebssystem wieder hochfährt, erzeugt es einen Arbeitsspeicherabzug unter "/var/crash", den Sie wieder löschen müssen.

Verhalten im Notfall bei MX500:

- AUTOBOOT ausschalten
- RESET drücken
- Dump (Hauptspeicherabzug) erzeugen und in slice 6 schreiben:

  *
  b 80 sp(0,0) sp(0,6)

- AUTOBOOT einschalten
- wieder normal hochfahren (siehe Kapitel 6)

Feststellen von Hardwarefehlern

Wie erkennt man Hardwarefehler ?

Hardwarefehler jeglicher Art machen sich in jedem Fall durch Systemmeldungen **an der Konsole** bemerkbar. Diese Meldungen werden im Mehrbenutzerbetrieb protokolliert:

⇨ Datei: "**/var/adm/klog.msg**"

Ein harmloser "**Hardwarefehler**" liegt beispielsweise vor, wenn Sie auf eine schreibgeschützte Diskette oder auf ein leeres Diskettenlaufwerk schreiben wollen.

Die Systemmeldungen enthalten immer das Kürzel des betroffenen SINIX-Treibers plus der Nummer der Plattensteuerung, der Platte, der Diskette oder des Streamers.

Beispiele für MX300 (vgl. L 3- 3)		**für MX500:**
hd0 oder hd1	1. bzw. 2. Platte	ssd (0,0), ssd(0,1)
fd0 oder fd 1	Diskettenlaufwerke	fd0, fd1
it0	Streamer (Magnetbandkassette)	ct0 oder ct1
exos0	1. Ethernet Baugruppe (LAN)	se0, se1
exa ncr0	Exabyte-Magnetband- kassettenlaufwerk	exa1t0 adp32.0
shd	externe SCSI-Platten	shd

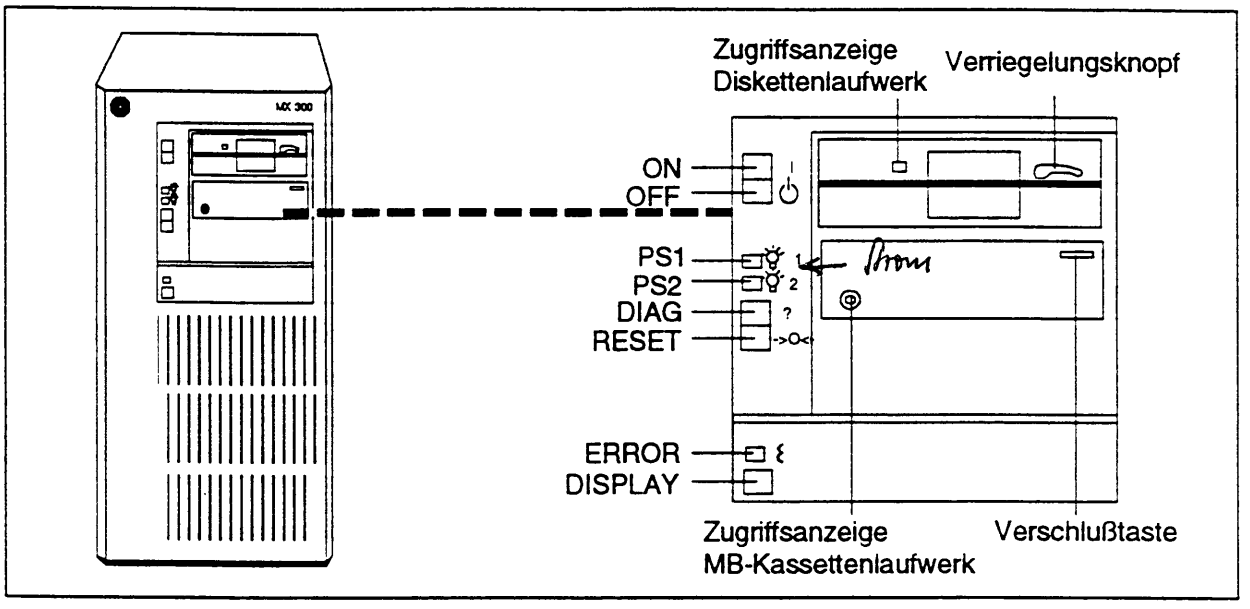
Im Anschluß an dieses Kürzel gibt der betroffene Treiber entweder eine eigene Meldung aus, oder er gibt die "**Fehlerbytes**" des Controllers in hexadezimaler Form weiter:

iopb : Fehlermeldung der Platten-/Disketten- und Streamer-Steuerung (Controller) in hexadezimaler Form (nur Storager)
Diese hexadezimalen Ausgaben sind wichtig für den Wartungstechniker !

Beispiel:

```
c0d0s11:HARD ERROR: blkno=1091426, disk sector (1372,12,50),...
iopb: 81 11 82 23 00 0C 05 5C 00 32 00 03 00 07 50 00 73 F8....
```

Bedienfeld MX300



Die Druckknöpfe der Bedieneinheit sind nur bei nach rechts gedrehtem Schlüssel funktionsfähig. Eine offene Frontklappe bedeutet noch nicht, daß der Schlüssel in der Stellung "offen" ist.

Druckknopf "ON"
Schaltet den MX 300 ein. Vorher sollte die Konsole eingeschaltet werden, da sonst der Rechner eventuell nicht hochfährt.

Druckknopf "OFF"
Niemals betätigen, sonst können Dateien verloren gehen oder zerstört werden!

Druckknopf "DIAG"
Dient zu Diagnosezwecken. Soll ein Systemdump gezogen werden, so ist erst der Druckknopf "DIAG" und dann der Druckknopf "RESET" zu drücken.

Druckknopf "RESET"
Ist nur in Fehlerfällen zu drücken, da Daten verlorengehen können.

Leuchtdiode "ERROR"
Leuchtet, wenn z.B. während des Einschaltvorgangs ein Hardwarefehler aufgetreten ist.

Zweistelliges Display
Betriebsanzeige. Leuchtet die gelbe Leuchtdiode auf, so wird hier der aufgetretene Fehler codiert angezeigt. Während SINIX läuft, wird zyklisch gezählt. Je nach Auslastung schneller oder langsamer (Idle-Lampe). Bei hoher Auslastung bleibt die Anzeige stehen.

Arbeitsspeicherabzüge

Zu Diagnose-Zwecken sichert der Rechner nach folgenden Ereignissen den Inhalt des kompletten Arbeitsspeichers und Zusatzinformationen im Verzeichnis "**/var/crash**".

- nach einem Systemabsturz oder DIAG+ RESET (sinnvoll)
- nach Stromausfall oder unsachgemäßem Abschalten des Rechners (Inhalt des Speichers leer)

Der Arbeitsspeicherabzug ist nur verwendbar, wenn der Rechner nach dem "Absturz" nicht ausgeschaltet wurde !!!

Der Rechner sichert den Arbeitsspeicher normalerweise in 2 Schritten:

1) physikalische Kopie in den "swap"-Bereich oder nach /dev/dump (slice 6)

Wurde das Betriebssystem nicht ordnungsgemäß heruntergefahren, kopiert der Rechner vor dem nächsten Systemstart den Inhalt des Arbeitsspeichers beim MX300 in den "**swap**"-Bereich der 1. Platte und beim MX500 in den slice, den Sie für /dev/dump vorgesehen haben (normalerweise slice 6).

2) Sicherung nach "/var/crash":

Beim Systemstart wird beim Übergang in den Mehrbenutzer-Betrieb automatisch ein Programm "**savecore**" aufgerufen, das den Arbeitsspeicherabzug aus dem "**swap**"oder "**/dev/dump**"-Bereich (wo er überschrieben würde) nach "**/var/crash**" kopiert.

Der Arbeitsspeicherabzug hat die Größe des Arbeitsspeichers und belegt daher viel Platz auf der Platte.

Nach jedem Stromausfall oder nach unsachgemäßem Abschalten des Systems müssen Sie den Arbeitsspeicherabzug löschen.

Aufgabe 4:

In Ihrem Dateiverzeichnis finden Sie die Datei **"klog.msg"**, die die Meldungen eines MX300-Kerns enthält.

Bitte stellen Sie den genauen Hardwareausbau dieses MX300 fest:

- Arbeitsspeicher
- Plattensteuerungen (Controller)
- Platten und Plattentypen, Diskettenlaufwerke
- E/A-Prozessoren
- DFÜ/LAN-Prozessoren

Wieviel Arbeitsspeicher benötigt der Systemkern ?

Von welcher Hardware stammen die Fehlermeldungen, die Sie in dieser Datei finden?

"run-level", init

In SINIX V5.4 sind mehrere Betriebsarten ("**run-level**") für das Betriebssystem definiert.
Für jede Betriebsart (jeden run-level) ist festgelegt, welche Prozesse in diesem run-level laufen dürfen:

run-level 0: Betriebssystem herunterfahren (keine Prozesse)

run-level 1: Einbenutzer-Betrieb: nur die Konsole wird bedient
Prozesse, die ohne Terminal oder an der Konsole laufen, werden jedoch nicht beendet ! Sie können solche Prozesse mit dem Kommando ***"killall 15"*** oder ***"killall 9"*** beenden.

run-level s,S: Einbenutzer-Betrieb mit Kennwortabfrage: die virtuelle Konsole "**syscon**" wird auf das Terminal gesetzt, an dem **"init s"** aufgerufen wurde. Hintergrundprozesse werden auch nicht beendet.

run-level 2: normaler Mehrbenutzer-Betrieb

run-level 3: Mehrbenutzer-Betrieb + NFS

run-level 6: herunterfahren und reboot

run-level a,b,c: zum Start zusätzlicher Anwendungen

run-level ändern *(Beispiele):*

a) `init s` # Wechsel in Einbenutzer-Betrieb
an beliebigem Terminal

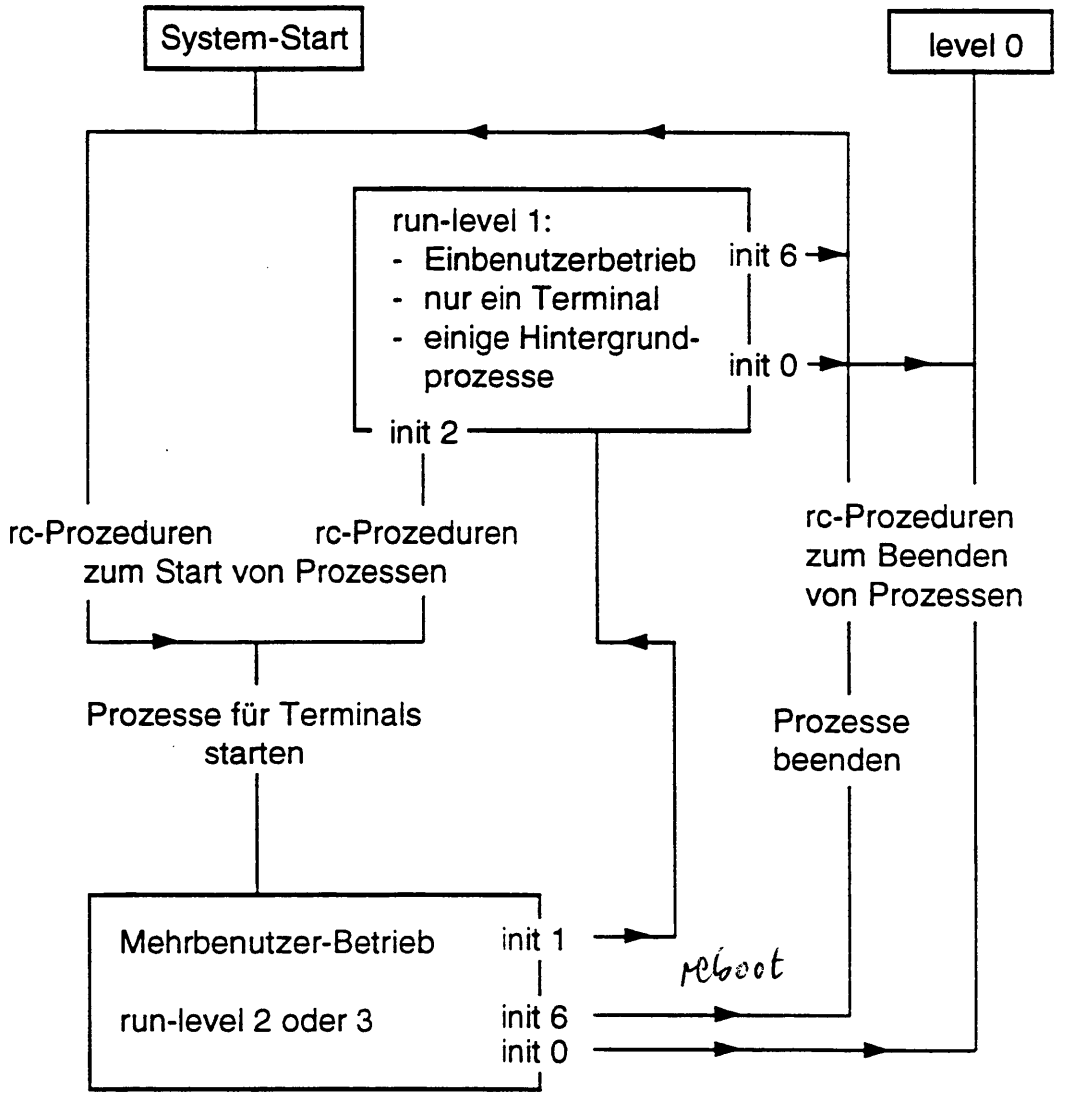
---

b)
```
sync; sync
init 0      # System schnell herunterfahren
            # (ohne Warnung der Benutzer)
```

Aktuellen run-level abfragen:

```
who -r
```

run level 0, 1, 2 und 3



Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

/etc/inittab

In der "inittab" sind alle Prozeduren und Programme eingetragen, die der "init" beim Übergang in einen anderen run-level automatisch starten soll. Sie können zu jeder Prozedur/jedem Programm die **Aktion** des "init" festlegen:

respawn	Prozeß im Hintergrund starten: falls sich der Prozeß beendet, Prozeß neu starten
wait	Prozeß starten und Prozeßende abwarten
once	Prozeß einmal starten, ohne das Prozeßende abzuwarten
sysinit	Prozeß nur beim Systemstart starten
off	Prozeß (falls vorhanden) mit SIGTERM und SIGKILL beenden
initdefault	in dieser Zeile steht der run-level, in den das Betriebssystem hochfährt

Jede Zeile besitzt 4 Felder, getrennt durch Doppelpunkt:

```
bchk::sysinit:/sbin/bcheckrc </dev/console >/dev/sysmsg 2>&1
is:2:initdefault:
r0:0:wait:/sbin/rc0 off 1> /dev/sysmsg 2>&1 </dev/console
r1:1:wait:/sbin/rc1 1> /dev/sysmsg 2>&1 </dev/console
r2:23:wait:/sbin/rc2 1> /dev/sysmsg 2>&1 </dev/console
r3:3:wait:/sbin/rc3 1> /dev/sysmsg 2>&1 </dev/console
r5:5:wait:/sbin/rc0 reboot 1> /dev/sysmsg 2>&1 </dev/console
r6:6:wait:/sbin/rc6 reboot 1> /dev/sysmsg 2>&1 </dev/console
sd:0:wait:/sbin/uadmin 2 3 >/dev/sysmsg 2>&1 </dev/console
fw:5:wait:/sbin/uadmin 2 2 >/dev/sysmsg 2>&1 </dev/console
rb:6:wait:/sbin/uadmin 2 1 >/dev/sysmsg 2>&1 </dev/console
sc:234:respawn:/usr/lib/saf/sac -t 300
sc1:23:off:/sbin/getty ttyc1 S
s000:12345:respawn:/sbin/getty term/tty000 S
s001:23:respawn:/sbin/getty term/tty001 S
s003:23:respawn:/sbin/getty term/tty003 S
```

Aktion

Shell-Kommando + Parameter

Verweis auf /etc/gettydefs

Liste der run-level oder leer

eindeutiger Identifikator pro Zeile (max. 4 Zeichen)

inittab verändern

Bei einer Neukonfiguration des Systemkerns (Kernel-Environment) wird die "/etc/inittab" ~~eventuell~~ neu erzeugt. Die "/etc/inittab" entsteht dabei aus folgenden Dateien:

/etc/conf/cf.d/init.base # Basis der inittab

/etc/conf/init.d/sc # Service-Konsole
/etc/conf/init.d/sr # Terminals
/etc/conf/init.d/... # ev. weitere Dateien

Eine dauerhafte Veränderung der Konfiguration sollten Sie daher auch in diesen Dateien vornehmen.

autopush, /etc/ap/chan.ap

Zu Beginn der Datei "/etc/inittab" finden Sie die Zeile:

```
ap::sysinit:/sbin/autopush -f /etc/ap/chan.ap
```

Die Datei "/etc/ap/chan.ap"

```
# /dev/console and /dev/contty  autopush setup
#
# major        minor   lastminor        modules

  1          -1        0              ldterm ttcompat
  38         0         96             ldterm ttcompat
  50         0         196            ldterm ttcompat
  36         0         255            ldterm ttcompat
```

Majornummer

erste Minor-nummer
falls -1: alle Minor-nummern

letzte Minornummer oder 0

Kernel-Module für "STREAMS"-Treiber

In UNIX System V.4 basieren die meisten Treiber im Systemkern auf "STREAMS". Solche "STREAMS"-Treiber sind weitgehend modular und konfigurierbar.

Mit dem Kommando "autopush" können Sie "STREAMS"-Treiber konfigurieren und vorhandene Kernel-Module "zwischenschalten".

ldterm Modul für die Standard-Linediscipline

ttcompat Berkeley und XENIX Kompatibilität für Linediscipline

Bei SINIX V5.40 sind jedoch noch nicht alle Treiber auf "STREAMS" umgestellt.

Kontrollieren, welche Module für einen Treiber konfiguriert sind:

```
autopush -g -M 36 -m 0    # get configuration
```

Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

Betriebssystem herunterfahren

Die Rechner MX300 und MX500 müssen Sie unbedingt softwaregesteuert unter der Benutzerkennung **"root"** mit der **"shutdown"**-Prozedur oder mit **"init 0"** herunterfahren. Der MX300 schaltet sich anschließend automatisch ab.

Sie dürfen den MX300 niemals über die Ausschalttaste abschalten !!

Die Prozedur **"shutdown"**:

shutdown [-y] [-gmm] [-i[0sS156]]

-y — keine weiteren Abfragen

-g[0-9]* — Zeit bis zum shutdown (Standard: 60 Sek.)

-i[0sS156] — neuer run-level (Standard: 0)

Um die **"shutdown"**-Prozedur aufzurufen, müssen Sie sich als **"root"** anmelden und in das "/" Verzeichnis wechseln.

Die Prozedur sendet eine Warnung an alle angemeldeten Benutzer.

Beispiele:

a) In 10 Minuten herunterfahren und ausschalten:

b) In 3 Minuten einen Reboot durchführen:

c) Sofort in den Einbenutzerbetrieb herunterfahren:

rc-Dateien in V5.4

Beim Übergang in einen neuen run-level werden viele kleine produktspezifische Prozeduren in den Dateiverzeichnissen "/etc/rc0.d", "/etc/rc1.d", ... gestartet:

run-leve	Prozedur + Parameter	Dateiverzeichnis mit Prozeduren
0	/sbin/rc0 off	/etc/rc0.d
1	/sbin/rc1	/etc/rc1.d
2	/sbin/rc2	/etc/rc2.d /etc/idrc.d
3	/sbin/rc2 /sbin/rc3	/etc/rc2.d,/etc/idrc.d /etc/rc3.d
5	/sbin/rc0 reboot	/etc/rc0.d
6	/sbin/rc6 reboot	/etc/rc0.d

Prozeduren in /etc/rc?.d:

SnnNAME — Prozeduren zum Starten von Prozessen

KnnNAME — Prozeduren zum Beenden (Kill) von Prozessen

└ Produkt-NAME (z.B. **"nfs"**, **"lpr"**, ...)

nn - Nummer bestimmt die Aufrufreihenfolge

Ein Verweis dieser Prozeduren steht jeweils im Verzeichnis /etc/init.d.

Prozeduren, deren Name mit **"K"** beginnt, werden mit dem Parameter "**stop**" aufgerufen, Prozeduren, deren Name mit **"S"** beginnt, mit dem Parameter "**start**".

Prozedur /etc/idrc.d/hd: (nur MX300)

Die Zugriffsrechte aller Plattengeräte aus der Datei "/etc/conf/node.d/hd" werden bei jedem Systemstart auf "600" gesetzt! Dies führt zu Fehlern bei Informix-Turbo bzw. Informix-ONLINE.

Aufgabe 5 (rc-Dateien)

a) Sehen Sie sich die Datei "/etc/inittab", die Prozedur "/sbin/bcheckrc" und die rc-Prozeduren in den Verzeichnissen "/etc/rc2.d" und "/etc/idrc.d" an.

Skizzieren Sie, was beim Hochfahren passiert. Welche Prozeduren werden in welcher Reihenfolge durchlaufen und wozu dienen die Prozeduren. Sehen Sie sich die Prozeduren bis einschließlich "S20klog" genauer an.
Sie müssen in den Prozeduren nur die Kommentare lesen.

b) Was müssen Sie beim MX300 genau tun, damit bei jedem Systemstart ein 2. swap-Bereich aktiviert wird?

Wie funktioniert das beim MX500? Sehen Sie sich gegebenfalls die entsprechende rc-Prozedur an.

Hinweise zu den Prozeduren "bcheckrc" und "mountall" bzw. "mountallp":

- fsck mit Schalter "-m" prüft nur den Superblock eines Dateisystems. Falls das System nicht ordnungsgemäß heruntergefahren wurde, liefert "fsck...-m..." einen Fehler und in den obigen Prozeduren wird automatisch ein fsck mit dem Schalter "-y" aufgerufen, der das Dateisystem repariert.

  Wurde das System ordnungsgemäß heruntergefahren, findet der fsck mit dem Schalter "-m" keine Fehler im Dateisystem.

- Das "root"-Dateisystem ist beim Hochfahren bereits montiert und wichtige Datenstrukturen des Dateisystems befinden sich in Arbeitsspeicher. Falls fsck einen Fehler im "root"-Dateisystem findet, sind 2 Maßnahmen notwendig:

1.) vor der Reparatur muß dafür gesorgt werden, daß die Datenstrukturen im Arbeitsspeicher nicht auf die Platte zurückgeschrieben werden:

```
/sbin/fsf off   # siehe /sbin/bckeckrc
```

2.) nach der Reparatur muß das root-Dateisystem neu montiert oder das System neu gestartet werden.

Fehler im Dateisystem

Bei jedem Systemstart werden alle Dateisysteme mit "**fsck**" überprüft:

- oberflächliche Prüfung aller Dateisysteme mit "**fsck -m**" (geht sehr schnell)
- die fehlerhaften Dateisysteme werden automatisch mit "**fsck -y**" repariert

Wie kann es zu Fehlern in den Dateisystemen kommen ?

1) Stromausfall oder unsachgemäßes Abschalten des Rechners
(nicht mit "**shutdown**" heruntergefahren)

   ⇨ kann zu gefährlichen Folgefehlern führen, wenn die Dateisysteme anschließend nicht sofort repariert werden

2) Hardwarefehler (Plattensteuerung oder Plattenfehler)
Hardwarefehler treten zunächst sehr sporadisch auf !

> Falls Sie an Ihrem Rechner Hardwarefehler vermuten, sollten Sie die Dateisysteme täglich prüfen (Ausbreitung von Fehlern vermeiden).

3) **Fehler des Systemverwalters:**

   z.B. fsck auf montierte Dateisysteme im Mehrbenutzer-Betrieb oder Konfigurationsfehler

4) unerkannte Fehler des SINIX-Systemkerns (sollte eigentlich nicht vorkommen !)

Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

Einbenutzerbetrieb: run-level 1

Der Systemverwalter fährt den Rechner in den Einbenutzerbetrieb, weil er bestimmte Aufgaben nicht im laufenden Betrieb durchführen darf:

- Datensicherung des Betriebssystems
- Überprüfung und Reparatur von Dateisystemen ("**fsck**")
- Dateisysteme umorganisieren (z.B. auf eine 2. Platte kopieren)
- ...

Der Systemverwalter ("**root**") hat 2 Möglichkeiten um an der Konsole in den "**run-level 1**" herunterzufahren:

1) `shutdown -y -g60 -i1`

2)
```
wall ...      # Benutzer warnen
init 1        # sofort und ohne Warnung herunterfahren
sleep 180     # ca. 180 Sekunden warten
killall 15    # allen Hintergrundprozessen Signal 15
              # senden
sleep 10      # ca. 10 Sekunden warten
killall 9     # allen Hintergrundprozessen Signal 9
              # senden
umountall     # Dateisysteme bis auf /, /stand und /proc abmontieren
```

Warum müssen Sie Hintergrundprozesse beenden, um Dateisysteme mit "**umountall**" abzumontieren ?

"fsck" selbst aufrufen ?

- **rufen Sie "fsck" auf, wenn Sie den Verdacht haben, daß Ihre Dateisysteme nicht in Ordnung sind**
- **"fsck" nur auf abmontierte Dateisysteme anwenden (vor "fsck" unbedingt "umount" bzw. "umountall") !!!**
- **"root"-Bereich nur im "run-level 1" überprüfen !**

- **als "root" an der Konsole anmelden**
- System in den Einbenutzerbetrieb heruntergefahren, Prozesse beenden, Dateisysteme abhängen (wie auf der letzten Seite).
- "fsck" nacheinander auf alle Dateisysteme, die in der /etc/vfstab stehen:

```
sync ; sync
fsck -y /dev/rdsk/c0d0s3     # "usr"-Dateisystem
fsck -y /dev/rdsk/c0d0s4     # "home"
fsck -y /dev/rdsk/c0d0s11    # "var"
fsck -y /dev/rdsk/c0d0s12    # "opt"
fsck -y /dev/rdsk/c0d0s13    # "tmp"
      ...
```

- Überprüfung des "root"-Dateisystems:

```
fsf off                      # Zurückschreiben der Puffer ausschalten
fsck -y /dev/rdsk/c0d0s1     # "root" Dateisystem
```

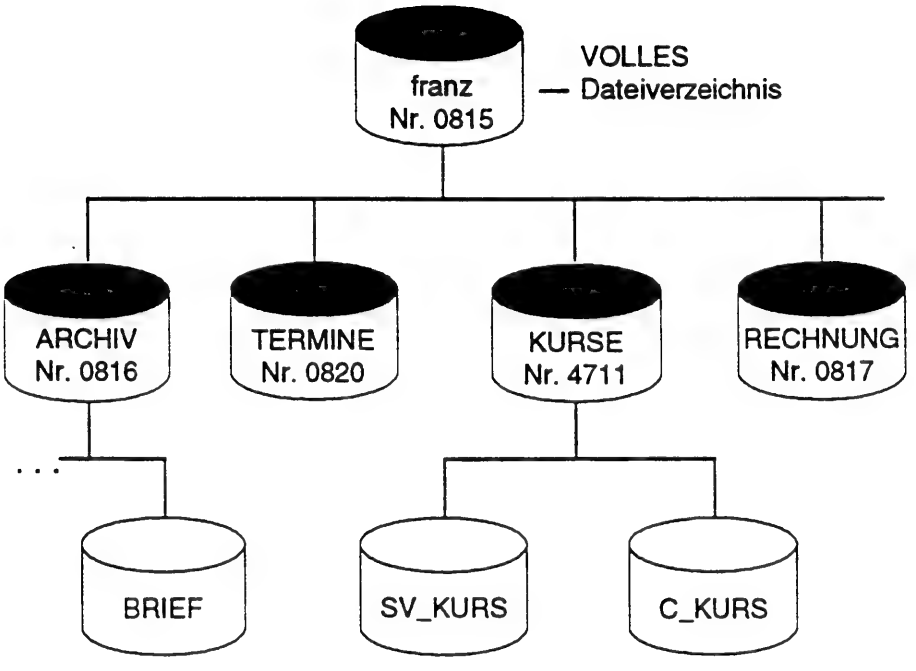
Falls Fehler aufgetreten sind, drücken Sie nun: **RESET**-Taste (ohne sync !!)
Andernfalls:

```
init 6                       # neu hochfahren
```

lost+found

"**fsck**" muß eventuell Dateien oder Dateiverzeichnisse "**wegreparieren**", um die Konsistenz des Dateisystems wieder herzustellen !

Was passiert, wenn "**fsck**" ein volles Dateiverzeichnis "**wegrepariert**" ?



Aufgabe 6:

Der Dozent kopiert ein defektes Dateisystem in Ihren Plattenbereich (slice). Überprüfen und reparieren Sie das Dateisystem und untersuchen Sie das "lost+found" Verzeichnis. Protokollieren Sie die fsck-Meldungen.

Regelmäßige Kontrollaufgaben

Protokolldateien kontrollieren und zurücksetzen:

In jedem UNIX-System werden aus Sicherheitsgründen zahlreiche Protokolldateien angelegt, die der Systemverwalter regelmäßig kontrollieren und zurücksetzen sollte. Andernfalls wachsen diese Protokolldateien immer mehr an und belegen viel Plattenplatz.

Sie können eine Protokolldatei auf 0 Bytes zurücksetzen, indem Sie einfach die Ausgabeumlenkung der Shell verwenden:

Beispiel: **> /var/adm/klog.msg**

Protokolldateien stehen in den Dateiverzeichnissen **"/var/adm"** und **"/var/cron"**.

Es existiert eine Prozedur **"/sbin/cleanup"**, die einige Protokolldateien automatisch zurücksetzt. Diese Prozedur sollten Sie erweitern (siehe Anhang).

Sie müssen unter anderem folgende Dateien regelmäßig zurücksetzen:

/var/adm/klog.msg	Meldungen des Systemkerns
/var/adm/lastlog	Datum der letzten Anmeldung für jeden Benutzer
/var/adm/sulog	Protokoll der "su"-Aufrufe zum Superuser
/var/adm/wtmp	Protokoll aller Anmeldungen lesen mit: who /var/adm/wtmp
/var/adm/wtmpx	**erweitertes Protokoll aller Anmeldungen (wird sehr groß!!)**
/var/adm/bootplog	Protokolldatei für bootp (X-Terminals)
/var/cron/log	Protokolldatei des "cron"
/var/crash/*	Arbeitsspeicherabzüge nach Systemabsturz oder Stromausfall (riesig groß !!!!)

In einigen Versionen von SINIX V5.4 werden die temporären Dateien unter "/tmp" nicht automatisch gelöscht !

Sie müssen eventuell auch unter "/tmp" aufräumen und alle Dateien löschen, die älter als 3 Tage sind.

Weitere Protokolldateien

"accounting"

Falls "accounting" eingeschaltet ist (/etc/rc2.d/S22acct) werden folgende Dateien angelegt:

/var/adm/pacct # eventuell: /var/adm/pacct?

Diese Dateien können Sie z.B. auswerten mit dem Kommando **"acctcom"** (siehe Kommandohandbuch).

Das gesamte "accounting "-System ist recht kompliziert. Falls Sie ONLINE-Manuale besitzen, erhalten Sie Informationen über:

```
man acctsh        # Informationen über accounting
```

Messdateien für Systemauslastung

Im Verzeichnis **"/var/adm/sa"** finden Sie Messdateien "sa??" und Auswertungen "sar??" des "sar"-Kommandos. In diesen Messdateien wird die Systemauslastung über den ganzen Tag mit Messintervallen von 20 Minuten festgehalten.

Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

Prozeduren automatisch starten

Kommandos, die regelmäßig zu bestimmten Zeiten gestartet werden sollen, können Sie in eine crontab-Datei eintragen (z.B. Datensicherung um 22.00 Uhr).
Dies ist besonders nützlich bei Rechnern, die auch nachts laufen.

crontab Kommando:

In SINIX V5.4 kann jeder berechtigte Benutzer mit dem "crontab"-Kommando eine eigene "crontab"-Datei anlegen:

```
vi mycrontab          # eigene crontab erstellen

crontab mycrontab     # eigene crontab "absenden"

crontab -l            # aktuelle crontab-Datei ansehen

crontab -r            # aktuelle crontab-Datei deaktivieren
```

crontab Dateien:

Prozeduren oder Kommandos zur Systemverwaltung sollten in der "root-crontab" stehen. Sie finden die "crontab"-Dateien der einzelnen Benutzer im Verzeichnis:

/var/spool/cron/crontabs/

Beispiel für eine "root"-crontab-Datei:

```
#ident "@(#)adm:root 1.5.1.2"
#ident "$Header: root 2.1 90/05/18 $"
10      15      * * 5    /bin/su root -c   "/sbin/cleanup > /dev/null"
48      11,14   * * 1-5  /bin/su uucp -c  "/usr/lib/uucp/uudemon.admin > /dev/null 2>&1"
40      *       * * *    /bin/su uucp -c  "/usr/lib/uucp/uudemon.poll > /dev/null"
26,56   *       * * *    /bin/su uucp -c  "/usr/lib/uucp/uudemon.hour > /dev/null"
0       2       * * 0,4  /etc/cron.d/logchecker
```

Minute(n) — Stunde(n) — Tag(e) im Monat — Monat im Jahr (1-12) — Tag in der Woche (0-6 oder 1-7, je nach System)

Die Prozedur "/sbin/cleanup" soll Aufräumarbeiten unter "/var/adm" durchführen! Sie müssen allerdings diese Prozedur an Ihrem System erweitern und zu einer Zeit starten, zu der Ihr System läuft!

Das "at"-Kommando:

- startet Kommandos oder Prozeduren zu einem späteren Zeitpunkt
- "**at**" liest die Kommandos oder die Prozedur von der Standardeingabe

Beispiele:

```
at 18:10 < prozedur     # startet prozedur nach 18:10 Uhr

at -l                   # listet alle offenen Aufträge auf
```

Achtung:

- crontab-Aufträge werden nur erledigt, wenn der Rechner zum angegebenen Zeitpunkt läuft
- "**at**"-Aufträge werden auch noch nachträglich ausgeführt

Berechtigung für crontab und at

Die Berechtigung zur Verwendung des "**crontab**"-Kommandos muß der Systemverwalter über eine der beiden folgenden Dateien festlegen:

/etc/cron.d/cron.deny oder alternativ:
/etc/cron.d/cron.allow

- Ist "**cron.allow**" vorhanden, so dürfen nur die Benutzer mit "**crontab**" arbeiten, deren Name in der Datei "**cron.allow**" steht.
- Ist "**cron.allow**" nicht vorhanden sondern "**cron.deny**", so dürfen alle Benutzer mit "**crontab**" arbeiten, deren Name nicht in der Datei "**cron.deny**" steht.

Berechtigung für "at"

Der Systemverwalter vergibt die Berechtigung für das "**at**"-Kommando über die Dateien:

/etc/cron.d/at.deny oder alternativ:
/etc/cron.d/at.allow

Prozeduren aus einer crontab starten :

- Prozeduren, die aus einer crontab gestartet werden, sollten ihre Umgebungsvariablen (z.B.: PATH) selbst setzen und exportieren:

  PATH=/sbin:/usr/bin:/usr/sbin:/opt/bin; export PATH

- Außerdem sollten Sie die Standard-Ausgabe- und Fehler-Känale auf eine Datei umlenken oder über Pipe als "mail" versenden. Dies gilt auch für "at"-Aufträge !

Systemzeit, Weltzeit, Hardwareuhr

Frage: Wie werden in UNIX Zeit und Datum gemessen?

In UNIX-Systemen unterscheidet man zwischen:

- der lokalen Zeit, die von den Kommandos (z.B. "date" oder "ls -l") angezeigt wird (gesteuert von der TZ-Variablen)
- der Systemzeit, die auf Weltzeit UTC (Universal Time Coordinated) bzw. GMT (Greenwich Mean Time) eingestellt sein sollte
- der Hardwareuhr, die mit der Systemzeit identisch sein sollte

TZ-Variable:

Sowohl Systemzeit als auch Hardwareuhr sollten immer auf Weltzeit eingestellt sein. Zur Umstellung auf Sommerzeit oder auf eine andere Zeitzone dient allein die **Variable TZ.** Das allgemeine Format der TZ-Variablen finden Sie im Kommandohandbuch bei der Beschreibung der Shell-Variablen (sh). Wir behandeln einige Beispiele für die TZ-Variable.

In SINIX V5.4 und vielen anderen UNIX-Systemen gibt es eine zentrale Datei, in der die TZ-Variable systemweit vorbelegt wird:

```
/etc/TIMEZONE
```

Middle european time: MET oder deutsch: MEZ

In der Bundesrepublik gilt normalerweise die mitteleuropäische Zeit, die sich um eine Stunde von der Weltzeit unterscheidet. Die TZ-Variable kann dann wie folgt gesetzt sein:

```
TZ=MET-1   oder      TZ=MEZ-1
```

Sommerzeit: MEST oder deutsch: MSZ

Wird in der Bundesrepublik auf "Sommerzeit" umgeschaltet, so unterscheidet sich unsere Zeit um 2 Stunden von der Weltzeit:

```
TZ=MEST-2 oder       TZ=MSZ-2
```

TZ-Variable

Die meisten UNIX-Systeme unterstützen eine automatische Umschaltung von Winterzeit auf Sommerzeit und zurück. Leider verwendet jedes System dabei eine andere Technik. Bei SINIX-ODT oder SCO-UNIX kann die TZ-Variable beispielweise wie folgt aussehen:

```
TZ=MEZ-1MSZ,M3.5,M9.5
           |    |    |
           |    |    Zurückschalten: 5.Woche/9.Monat
           |    Umschaltung:  5.Woche des 3.Monats
Name der Sommerzeitzone
```

Einige Standard-Formatelemente der TZ-Variablen:

- Das TZ-Format beginnt mit dem Namen der Zeitzone. Dieser Name ist standardisiert, aber auch neu definierbar:

  ```
  TZ=BAYERN-2
  ```

- Im Anschluß an den Namen folgt oft ein Minus- oder ein Plus-Zeichen und eine Ziffernfolge:
  -2 bedeutet: 2 Stunden vor (östlich) GMT
  +1 bedeutet: 1 Stunde nach (westlicht) GMT

- Steht im Anschluß an die Ziffernfolge ein weiterer Name für eine Zeitzone, stellt UNIX automatisch auf Sommerzeit um. Die Regeln für die Umstellung können dann im Anschluß (durch Kommata getrennt) definiert werden.

Spezielles Format in SINIX V5.4:

In SINIX V5.4 gibt es (wahlweise) eine andere Technik:

```
TZ=:MET     # in /etc/TIMEZONE eintragen

cd /usr/lib/locale/TZ
/usr/sbin/zic europe
```

Das Kommando "zic europe" erzeugt im aktuellen Verzeichnis eine binäre Datei "MET", in der die Regeln zur Zeitzone und zur Sommerzeit hinterlegt sind.
Der Doppelpunkt in der TZ-Variablen sorgt dafür, daß nach der Datei "/usr/lib/locale/TZ/MET" gesucht wird.
Bei dieser Methode können Sie den Namen der Zeitzone nicht mehr frei wählen, da eine Datei dieses Namens im Verzeichnis "/usr/lib/locale/TZ" vorhanden sein muß:

```
TZ=:BAYERN      # Fehler !!!!!
```

Systemzeit und Hardwareuhr stellen

Stellen Sie die Uhrzeit niemals im laufenden Betrieb zurück! Wechseln Sie vorher in den Einbenutzerbetrieb und beenden Sie alle Prozesse.

- Beim MX500 können Sie ganz herunterfahren und die Hardwareuhr im Hardwaremonitor auf die GMT-Zeit einstellen.

- Beim MX300 verwenden Sie das "date"-Kommando, das sowohl die Systemzeit als auch die Hardwareuhr setzt. Achten Sie darauf, daß Ihre TZ-Variable richtig gesetzt ist:

```
echo $TZ            #   TZ-Variable ok ??

date MMddhhmmYY     #   MM == Monat, YY = Jahr
                    #   das Jahr muß angegeben werden!
```

  Sie geben dabei nicht die GMT-Zeit sondern die lokale Zeit an, das "date"-Kommando wertet die TZ-Variable aus und wandelt Ihre Zeitangabe in GMT-Zeit um.

  Die Option "date -u", mit der Sie direkt die GMT-Zeit eingeben könnten, funktioniert momentan nicht korrekt.

- In einem LAN-Netz müssen Sie die Zeit am Netzverwaltungsrechner umstellen, da normalerweise alle anderen Rechner ihre Systemzeit an den Netzverwalter anpassen.

- Bei vielen UNIX-Systemen setzt das "date"-Kommando nur die Systemzeit, die Hardwareuhr müssen Sie dann über ein anderes Kommando setzen.

Datensicherung

Geräte zur Datensicherung

1/4 Zoll Magnetbandkassettenlaufwerke (Streamer):

- **altes 60 MB Laufwerk mit 9 Spuren: TDC 3319 / TDC 3610**

- **155 MB Laufwerke mit 18 Spuren: TDC 3650 oder TDC 3660**

  Die 155 MB Laufwerke sind seit 09.89 wahlweise in alle neuen MX300, MX500 und WX200 Modelle eingebaut.
  Die Laufwerke arbeiten mit einer größeren Schreibdichte, daher können Sie mit diesen Laufwerken nur noch 60/155 MB Kassetten U7-H3 beschreiben.

Die 155 MB-Laufwerke können Bänder lesen, die mit 60 MB-Laufwerken beschrieben wurden.

Beschreiben Sie jedoch ein Band mit einem 155MB Laufwerk, so können Sie dieses Band auf keinem Rechner einlesen, der noch ein 60 MB-Laufwerk besitzt !!!

Magnetbandkassettenlaufwerk mit 2.0 GB (Exabyte):

Externes Gerät, das mit Video 8 (mm) Kassetten und rotierenden Schreib-/Lese-Köpfen arbeitet (Exabyte).

Für dieses Gerät benötigen Sie einen SCSI-Controller (Hostadapter) vom Typ 1, der in Ihren Rechner eingebaut sein muß. An diesen SCSI-Controller können Sie jedoch keine SCSI-Platten anschließen !

Sicherung auf externe SCSI-Platten:

Für externe SCSI-Platten (760 MB) benötigen Sie einen SCSI-Controller vom Typ 2 und einen Zusatzschrank.

Arbeiten mit Magnetbandkassette

Gerätedateien für MB-Kassetten:

a) Automatisches Zurückspulen beim "**close**" auf das Gerät:

Streamer: /dev/tape oder /dev/rmt/c0s0

Exabyte: /dev/exa0 oder /dev/rmt/exa?t0

b) Das Band wird nicht automatisch zurückgespult:

Streamer: /dev/tapen (norewind) oder /dev/rmt/c0s0n

Exabyte: /dev/exa8 oder /dev/rmt/exa?t0n

Wird das Streamer-Laufwerk geöffnet, so spult der Treiber das Band vor dem nächsten Zugriff automatisch zurück.

Standard-Streamer zurückspulen und straffen:

Straffen bedeutet, das Band bis zum Ende vor- und zurückzuspulen (Dauer: ca. 2 Minuten). Dieser Vorgang ist wichtig für die Sicherheit des späteren Lese- oder Schreibvorgangs (siehe "**mt**": Bandkontrolle).

Falls Sie beim Einlesen eines Bandes Lesefehler erhalten, legen Sie das Band erneut ein, lassen das Band erneut straffen und zurückspulen und versuchen es noch einmal (siehe Seite L 4 - 6).

Datensicherung

Pufferung

In SINIX V5.x puffert der Systemkern jede Streamer-Ein-/Ausgabe automatisch mit einem 1 MB-Puffer und vermeidet damit einen langsamen Start-/Stop-Betrieb.

Blockungsfaktor:

Viele Sicherungskommandos besitzen Optionen, mit denen Sie die Größe der Datenblöcke angeben können, die das Kommando bei Ein- bzw. Ausgabe verwendet:

Beispiele:

tar :	Option	-b		
dd :	Option	bs=		
cpio:	Optionen	-C	bzw.	-B

Die Ein-/Ausgaben auf den Streamer werden in SINIX V5.x vom Systemkern gepuffert, so daß Sie normalerweise keinen "Blockungsfaktor" angeben müssen.

Bandendeerkennung:

Die automatische Pufferung des Systemkerns hat einen Nachteil:

Kein Kommando ist in der Lage, automatisch das Bandende festzustellen (auch nicht "cpio") !

Warum nicht ?

Einige Kommandos besitzen Optionen, mit denen Sie dem Kommando bereits bei Aufruf die Bandkapazität mitteilen können.

Beim Exabyte können Sie zwischen gepufferter und ungegepufferter Ein-/Ausgabe wählen: /dev/rmt/exa?t0v ungepuffert

Bei der **ungepufferten Ein-/Ausgabe** müssen Sie einen "Blockungsfaktor" angeben: cpio -C...
Die maximale Blockgröße des "privaten" Puffers ist bei SINIX V5.4 allerdings auf 32 KB begrenzt.

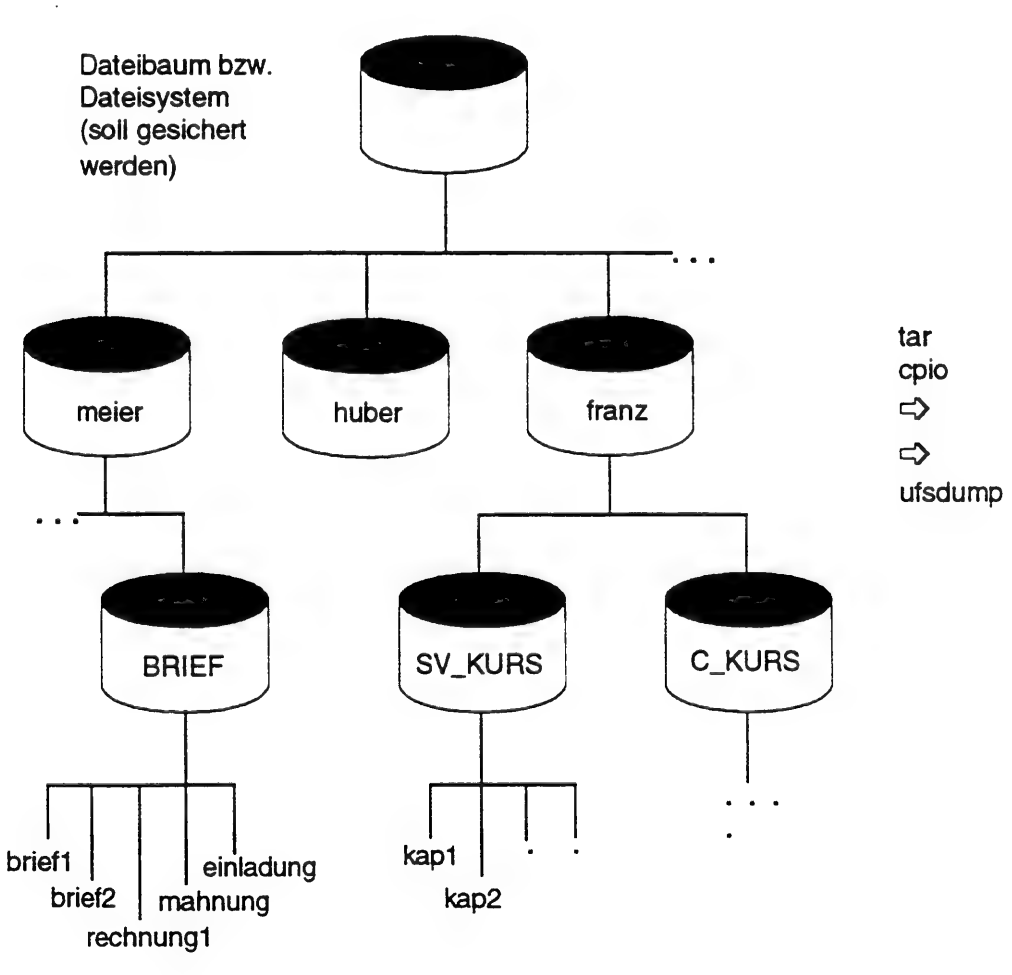
Grundprinzip von tar, cpio und ufsdump

Die Programme "**tar**", "**cpio**" und "**ufsdump**" dienen unter anderem zur Sicherung von Daten auf einem Magnetband.

tar, cpio **sichern beliebige Dateibäume**
oder einzelne Dateien

ufsdump **sichert komplette "ufs"-Dateisysteme (nicht "s5" oder"bfs")**
⇨ Einlesen mit: **ufsrestore**

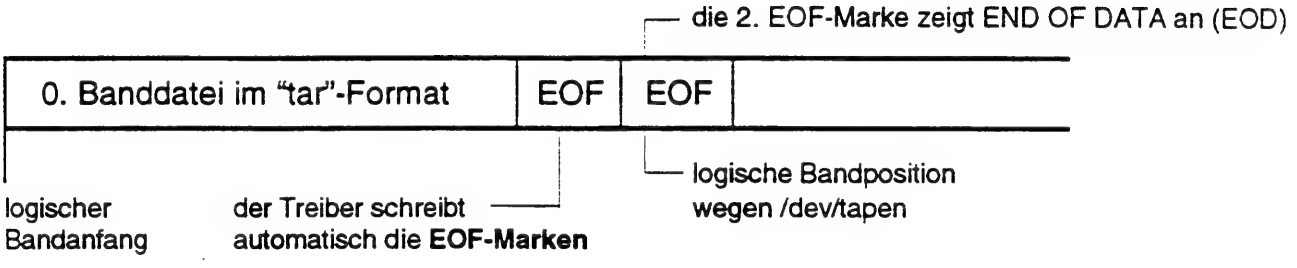
Was machen diese Programme mit den Daten ?



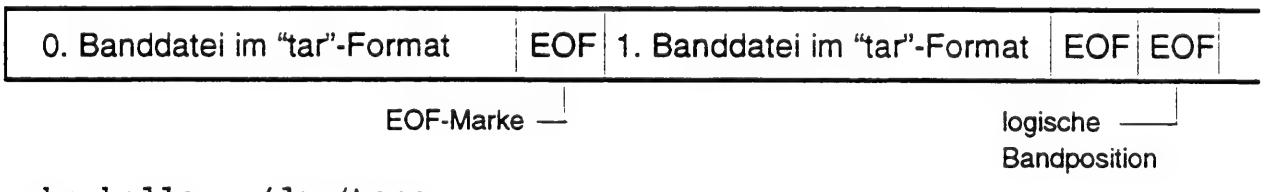
Erzeugen mehrerer Banddateien

Beispiel:

```
cd /home/meier
tar cvf /dev/tapen .      # tar erzeugt eine große Datei
                          # und schreibt sie auf das Band
```




```
cd /home1/mueller
tar cvf /dev/tapen .
```




```
echo hallo > /dev/tape
```

Was passiert mit den Daten, wenn Sie jetzt eingeben :

```
echo Unsinn >/dev/tapen ???
```

"mt": Bandkontrolle

Sie sollten bei der Installation des Betriebssystems auch das Paket "compat" (compatibility package) installieren. Sie können das Paket auch nachträglich installieren.

Wenn "compat" installiert ist, haben Sie unter "/usr/ucb" viele nützliche "Berkeley"-Kommandos zur Verfügung, unter anderem das Kommando "mt":

```
/usr/ucb/mt -f /dev/tape noret      # Straffen vermeiden

/usr/ucb/mt -f /dev/tape ret        # Magnetband straffen
```

Löschen/ Positionieren

```
/usr/ucb/mt -f /dev/tape erase      # das ganze Band löschen
                                    # (mit Löschkopf: 2 Min, bei
                                      Exabyte: Stunden)
echo Unsinn > /dev/tape             # ab der aktuellen
                                    # Stelle löschen

/usr/ucb/mt -f /dev/tapen fsf       # um eine Banddatei
                                    # vorwärts spulen

/usr/ucb/mt -f /dev/tapen fsf 2     # um zwei Banddateien
                                    # vorwärts spulen

< /dev/tape                         # an Bandanfang zurückspulen
```

Zurückpositionieren:

Mit dem Standard-Streamer ist es nicht möglich, um eine Banddatei zurückzupositionieren !

Wollen Sie mit dem 2.0 GB Magnetband-Kassetten-Laufwerk (Exabyte) um **eine Banddatei zurückspulen,** müssen Sie wie folgt vorgehen:

2 EOF-Marken zurück : `/usr/ucb/mt -f /dev/exa8 bsf 2`
1 EOF-Marke vorwärts: `/usr/ucb/mt -f /dev/exa8 fsf 1`

Sie können ähnliche Funktionen auch mit dem Kommando **"tapecntl"** ausführen.

Physikalische Einteilung des Standard-Streamerbandes

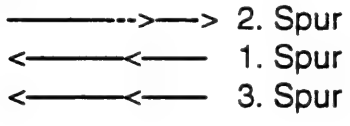
Physikalisch beschreiben die Standard-Streamer-Laufwerke nacheinander mehrere Spuren auf dem Magnetband. Jede Spur wird seriell beschrieben bzw. gelesen.

Das Magnetband wird dabei abwechselnd in beiden Richtungen transportiert.

> Die physikalische Bandposition stimmt daher nicht mit der logischen Bandposition überein.

Beispielschema :

Streamerband



Bänder kopieren ?

Da eine Magnetbandkassette sehr viele Banddateien enthalten kann, ist es nicht einfach, ein Band zu kopieren. Mit dem Kommandos "cp" oder "dd" können Sie jeweils nur eine Banddatei kopieren !

Beispiel:

```
cp /dev/tapen band.0        # kopiert die 0. Banddatei auf
                            # die Platte
cp /dev/tapen band.1        # kopiert die 1. Banddatei auf
                            # die Platte
    ...
```

Wenn Sie genügend Platz auf Ihrer Platte haben, können Sie zum Kopieren von Bändern folgende Prozeduren verwenden:

Beispiel:

Prozedur "cpin": von Band auf Platte

```
#!/bin/sh
# Allgemeine Prozedur zum Kopieren von
# Streamer-Band auf Platte: cpin
# (mit "cpout" wieder herausschreiben)
#
# Band straffen und zurueckspulen
/usr/ucb/mt -f /dev/tape ret
#
# 0. Banddatei kopieren
cp /dev/tapen band.0

#
# Alle anderen Banddateien kopieren
i=0
while test -s band.$i
do
    ls -l band.$i
    i=`expr $i + 1`
    if cp /dev/tapen band.$i
    then :
    else
        echo "Fehler oder Bandende !!"
# am Bandende kommen ev. Fehlermeldungen!
        break
    fi
done

# Band zurueckspulen
/usr/ucb/mt -f /dev/tape rewind
```

Prozedur "cpout": von Platte auf Band

```
#!/bin/sh
# Allgemeine Prozedur zum Kopieren von
# Platte auf Band: cpout
# (Dateien vorher mit "cpin" einlesen !)
#
# Band straffen und zurueckspulen
 /usr/ucb/mt -f /dev/tape ret
#


#
# Alle  Banddateien kopieren
i=0
while test -s band.$i
do
    ls -l band.$i
    if cp band.$i /dev/tapen
    then :
    else
        echo "Fehler beim Kopieren !!"
        exit 1 # bei Fehler abbrechen
    fi
    i=`expr $i + 1`
done
#
#
# Band zurueckspulen
/usr/ucb/mt -f /dev/tape rewind
```

Datensicherung

Sicherungskommandos

Kommando	Was kann gesichert werden ?	Geräte/ symbol. Links ?	Differenz-Sicherung?	Berücksichtigung der Bandlänge ?
tar	Dateibäume von "s5", "ufs" und "bfs"	nur in SINIX V5.4 !	mit "find ... -newer .." möglich	mit Schalter -k nnn (nnn in KB)
cpio mit find	Dateibäume von "s5", "ufs" und "bfs"	ja mit Schalter -c	mit "find ... -newer .." möglich !	Bandendeerkennung funktioniert nicht wegen gepufferter Ausgabe
dd und cp	physikalische Sicherung von beliebigen "slices"	ja	nein	nein
ufsdump ufsrestore	nur "ufs"-Dateisysteme	ja	eingebaute Funktion	ja (Folgebänder)

In SINIX V5.4 bietet Ihnen das Produkt **"DSX"** ein menügeführtes Datensicherungssystem, zur Sicherung von Benutzerdaten.

Der Nachteil komplexer, menügeführter Sicherungssysteme besteht jedoch darin, daß im **"Ernstfall"** (Datenverluste, defekte Platte, defekte Dateisysteme) zunächst das Sicherungssystem restauriert werden muß.

Wir wollen auch die Möglichkeiten zur Datensicherung betrachten, mit denen wir das System nach einem **"Ernstfall"** notfalls mit den **"boot"**-Disketten oder mit dem Systemband wieder restaurieren können.

cpio

1) **Dateibaum kopieren: cpio -p**

Beispiel: /home/meier —> /home1/mueller

```
cd /home/meier      # Quelle        Ziel:
find . -print       | cpio -pdm /home1/mueller
```

2) **Dateibäume sichern: cpio -o**

```
cd /home/meier      # Quelle

find . -print       | cpio -oVBc > /dev/tape
```

Wollen Sie Dateien mit "cpio" kopieren oder sichern, müssen Sie die Dateinamen über die Standardeingabe an "cpio" übergeben.

3) **cpio-Sicherungen einlesen: cpio -i**

```
cd /home1/mueller

cpio -ivdmu ['MUSTER'] < /dev/tape
```

Inhaltsverzeichnis: `cpio -itv < /dev/tape`

Bedeutung der verwendeten Schalter:

- B Blockgröße 5120 Bytes (Standard sind 512 Bytes)
- v Dateinamen und Meldungen ausgeben
- V für jede Datei einen Punkt ausgeben
- d notwendige Dateiverzeichnisse anlegen
- m das Änderungsdatum bleibt erhalten
- u neuere Dateien beim Einlesen überschreiben
- c "portables" Format (ab V5.4: Prüfsummen, Pfadnamen bis 1024 Bytes, nicht kompatibel zu älteren UNIX-Versionen)
- **H odc** entspricht dem "-c"-Format älterer UNIX Versionen (z.B.: SINIX V5.22)

Dateibaum kopieren

Teile des Dateibaums so kopieren, daß die Zugriffsrechte und die Modifikationszeiten der Dateien gleich bleiben.

Beispiel:

Sie wollen den Benutzer "**meier**", der bisher im "**/home**"-Dateisystem arbeitet, in das Dateisystem "**/home1**" umsiedeln. Der Benutzer "**meier**" soll von dieser Aktion möglichst nichts merken (keine Nachteile haben).

⇨ **Achtung**

In SINIX V5.2x ist "**cpio**" nicht geeignet um den "**root**" oder den "**usr**"-Bereich zu sichern oder zu kopieren !

In SINIX V5.4 können Sie "**cpio**" für alle Zwecke verwenden!

Gerätedateien für Disketten in V5.4

1. Laufwerk: 3,5 Zoll (Standard)

- high density : 1,4 MB Kapazität

AT&T-Gerätedatei : /dev/rdsk/**f03ht** | ohne Zylinder 0: f03h

XENIX-Gerätedatei : /dev/**rfd0135ds18** (135 tpi, 18 Sekt./Spur)

- double density : 720 KB Kapazität

AT&T-Gerätedatei : /dev/rdsk/**f03dt** | ohne Zylinder 0: f03d

2. Laufwerk: 5 1/4 Zoll (optional)

- high density : 1,2 MB Kapazität

AT&T-Gerätedatei : /dev/rdsk/**f15ht** | ohne Zylinder 0: f15h

XENIX-Gerätedatei : /dev/**rfd196ds15** (96 tpi, 15 Sektoren/Spur)

fl2-Bereich von SINIX V5.2x: /dev/**rdsk/tffl2h** (nur lesen)

- quad density : 720 KB Kapazität (double density mit 96 tpi)

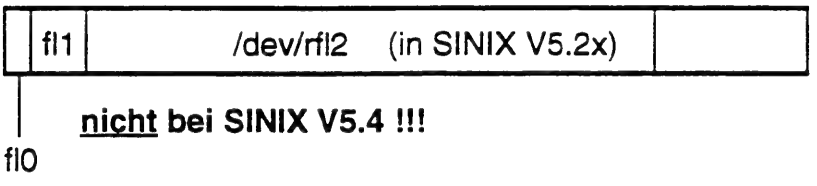
AT&T-Gerätedatei : /dev/rdsk/**f15qt** | ohne Zylinder 0: f15q

fl2-Bereich von SINIX V5.2x: /dev/rdsk/**tffl2** (nur lesen)

Gerätedateien, die das Format automatisch erkennen:	
1. Laufwerk : /dev/rdsk/**f0t**	ohne Zylinder 0: /dev/rdsk/f0
2. Laufwerk : /dev/rdsk/**f1t**	ohne Zylinder 0: /dev/rdsk/f1

Datenaustausch mit SINIX V5.2x

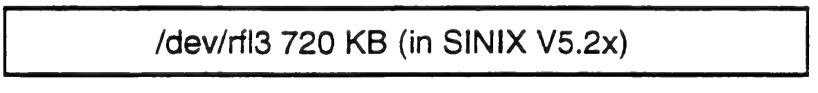
SINIX V5.4 unterstützt nicht mehr das ehemalige SINIX-Disketten-Format mit den Bereichen:

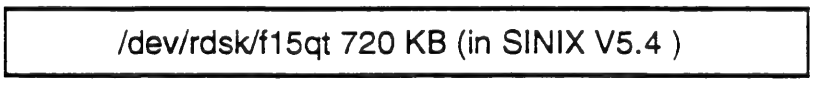


Falls Sie das optionale 5 1/4 Zoll Laufwerk besitzen, können Sie den "fl2"-Bereich von V5.2x-Disketten lesen aber nicht beschreiben:

/dev/rdsk/tffl2 bei Standard-Disketten
/dev/rdsk/tffl2h bei high density Disketten

Der Datenaustausch über das optionale 5 1/4 Zoll Laufwerk funktioniert am besten, wenn Sie das 720 KB Format und "double density" Disketten mit 96 tpi verwenden:





5 1/4 Zoll:	SINIX/UNIX V5.4	SINIX V5.2x
Formatieren:	format /dev/rdsk/f15qt	/etc/flformat -s 9 -b 512 -f N
Schreiben:	tar cvf /dev/rdsk/f15qt ...	tar cvf /dev/rfl3 ...
Lesen:	tar xvf /dev/rdsk/f15qt	att tar xvf /dev/rfl3 # Problem bei SINIX V5.21

Sie können statt "tar" auch das "cpio"-Kommando zum Datenaustausch verwenden (jedoch nicht mit Schalter "-c" !)

Wenn Ihr SINIX V5.2x Rechner ein 150 MB Streamerlaufwerk besitzt, tauschen Sie Ihre Daten am besten über Magnetbandkassetten aus.

Das tar Kommando

"**tar**" archiviert Dateien oder Dateibäume in einer "**tar**"-Datei. Die Datei kann auf Diskette, auf einem Magnetband oder auf der Platte liegen.

Unterschiedliche tar-Versionen

SINIX V5.2x		SINIX V5.4
"ucb"-tar	ab SINIX V5.22: neuer "att"-tar	ganz neuer "tar"
X/Open P. Guide 2 (XPG 2)	X/Open P. Guide 3 (XPG 3)	X/Open P. Guide 3 (XPG 3)
+ SINIX-Erweiterungen	ohne Erweiterungen	+ SINIX-Erweiterungen
		+ XENIX-Erweiterungen
sichert Pfadnamen bis 100 Bytes	sichert Pfadnamen bis 255 Bytes	sichert Pfadnamen bis 255 Bytes
keine Gerätedateien	sichert Gerätedateien	sichert Gerätedateien
steht auch im "sie"-Universum zur Verfügung		
"sie far" bzw. "ftar" für Disketten	ruft bei XPG2-Format automatisch den "ucb"-tar auf	kann XPG2-Format lesen

Die verschiedenen "**tar**"-Kommandos unterscheiden sich in vielen Punkten, u.a. in der Bedeutung der Schalter "**p**", "**o**" und "**k**".

Datensicherung

tar Aufruf:

Daten sichern (auf Band / Diskette schreiben) :

```
cd /home
tar cvf /dev/tape
```

Liste von Dateien / DVZ

```
mueller meier franz
```

schreiben — Streamer

auf Diskette:

```
tar cvfb /dev/rdsk/f0t  20  mueller meier franz
```

"Blockungsfaktor", Pufferung vom tar (Maximalwert: 20)

⇨ Geschwindigkeit

hinten anhängen (nur bei Disketten):

```
tar rvnfb /dev/rdsk/f0t  20 huber
```

Falls Sie nicht sicher sind, ob Ihre Daten auf das Band / die Diskette passen:

```
tar cvkf 150000 /dev/tape .
```

Bandlänge in KB — aktuelles DVZ

Kombination aller Schalter:

```
tar cvkfb 1440 /dev/rdsk/f0t  20  meier
```

Die Datei "/etc/default/tar"

Schreibarbeit sparen, durch Datei:

/etc/default/tar

Sie müssen in diese Datei sinnvolle Werte eintragen:

Beispiel:

	block	Länge
archive0=/dev/rdsk/f0t	20	1440
archive1=/dev/rdsk/f1t	20	1200
archive2=/dev/rdsk/tffl2	20	584
archive3=/dev/rdsk/f15qt	20	720
archive4=/dev/tape	20	150000

tar Aufruf mit Bezug auf /etc/default/tar

```
tar cv0   meier mueller franz

tar xv2   # einlesen von tffl2!

tar rvn0 huber
```

anhängen!

⇨ Achtung:

Falls eine Umgebungsvariable TAPE gesetzt ist, verwendet "tar" den Inhalt der Variablen als Gerät. In diesem Fall liest "tar" nicht die /etc/default/tar.

Datensicherung

tar

Vor- und Nachteile von "tar"

+ **einfache Bedienung:** Sie müssen keine Liste von Dateinamen über "**find**" erzeugen

Beispiel: Sie wollen "**/home/meier**" und "**/home/mueller**" sichern.
Wie lauten die Kommandos für "**tar**", wie für "**cpio**"?

+ **Berücksichtigung der Bandlänge** über den Schalter "-k"
(eventuelle Folgebandverarbeitung, demnächst auch bei "cpio")

- **"boot"-Disketten enthalten nicht "tar" sondern "cpio"**

- Die Liste der Dateinamen, die Sie als Argument übergeben können, ist systembedingt auf ca. 5 KB begrenzt (siehe Systemparameter ARG_MAX, L 5 - 18)
Dies führt zu Problemen bei Differenzsicherungen.
Warum ?

Deshalb: Spezielle Option "-I" bei SINIX V5.4:

```
tar cvf /dev/tape -I list_dat
```

Mit dieser Option sind auch umfangreiche Differenzsicherungen möglich.

Datensicherung

dd

SINIX alt → SINIX neu

Das Kommando **"dd"** ist ein Kopierprogramm (ähnlich "cp"), mit dem Sie einzelne Dateien oder komplette "slices" aus UNIX-Sicht physikalisch kopieren können.

Sie können z.B. "dd" benutzen, um:

a) die Dateisysteme "/", "/stand" und "/usr" in gleich große "slices" einer anderen Platte zu kopieren, damit im Notfall ein Ersatzsystem zur Verfügung steht

b) "slices" physikalisch auf Magnetbandkassette zu schreiben

c) Dateiinhalte oder Teile von Dateien ausgeben zu lassen

Einige Optionen von "dd":

if=ein_dat	Eingabedatei (Standard: Standardeingabe)
of=aus_dat	Ausgabedatei (Standard: Standardausgabe)
bs=n	Blockgröße für Eingabe- und Ausgabeblöcke
skip=n	die ersten n Eingabeblöcke überspringen
seek=n	in der Ausgabedatei n Blöcke überspringen
count=n	nur n Blöcke kopieren (Std: bis Dateiende)

"dd"-Aufrufe für die obigen Beispiele:

a) Vorbereitung: ???

Kopieren des "/"-Bereichs:

b) wie a) aber: of=/dev/tape

c) `dd if=/etc/passwd bs=128 count=1`

Datensicherung

Sicherungskonzept

Sicherung der Benutzerdaten:

Dateisysteme: /home, /home1, ...

Die Benutzerdaten können Sie mit beliebigen Kommandos (z.B. "tar" oder "ufsdump") sichern. Für Benutzerdaten bieten sich auch menügeführte Systeme an, wie z.B. "DSX".

Sicherung des Betriebssystems:

Dateisysteme: /stand, /, /usr, /var, /opt

Das Betriebssystem muß so gesichert werden, daß Sie es im Notfall (z.B. Ausfall einer Platte) von den Boot-Disketten oder vom Minirootsystem einlesen können. Sie müssen sich daher an folgende Regeln halten:

- Sicherungen mit "cpio -oBc" und zwar mit **relativen** Pfadnamen !

- Wichtig: Vor der Sicherung in den Einbenutzer-Betrieb wechseln und /dev/fd und /proc abmontieren !

- Die Platteneinteilungen (Slice-Größen) müssen separat gesichert werden, damit Sie eine Ersatzplatte neu einteilen können (prtutoc -f ....).

- Bereinigen Sie vor der Sicherung die Dateiverzeichnisse "/var/crash", "/var/adm", "/var/cron", ...

- Sie müssen vorher prüfen, ob das gesamte System auf ein Band paßt. Andernfalls müssen Sie z.B. "/opt" auf einem 2. Band sichern.

  Typischer Platzbedarf der Dateisysteme:

stand	1,3	MB	
root	9,0	MB	
usr	50 - 60	MB	
var	5,0	MB	(vorher bereinigen !!)
opt	50 - 100	MB	
Summe:	**116- 176**	**MB**	

Einem 155 MB-Band können Sie ca. 135-140 MB Nettodaten anvertrauen. Der Platzbedarf läßt sich nicht genau bestimmen, da Dateien mit mehreren "Hardlinks" vom "cpio" mehrfach gesichert werden.

Datensicherung

Arbeitsschritte zur Systemsicherung:

- in run level 1 herunterfahren

- alle Hintergrundprozesse beenden

- "/var/crash", "/var/adm" und "/var/cron" bereinigen (z.B. mit eigener Prozedur "cleanup")

- Platteneinteilungen mit "prtvtoc -f vtoc1 /dev/rdsk/..." lesen und mit "/etc/vfstab" vergleichen.
  Am besten erzeugen Sie für jede Platte eine Datei, die als Eingabedatei für "disksetup -ld ..." dienen kann. In Anhang des Kursordners finden Sie eine Prozedur "save_vtoc", die solche Dateien automatisch erzeugt.

  Die Ausgabedateien von "prtvtoc -f ..." und die Dateien, die als Eingabedatei für "disksetup" geeignet sind, sichern Sie mit "cpio" in die **0. Banddatei.**

- cd /
  Dateisysteme "/dev/fd" und "/proc" abmontieren

- "/" und "/stand" mit "cpio" und relativen Pfadnamen in **1. Banddatei** sichern

- "/usr" und "/var" mit "cpio" und relativen Pfadnamen in **2. Banddatei** sichern

- Falls "/opt" nicht mehr als 60 MB belegt, können Sie "/opt" in die **3. Banddatei** sichern

- Sie können die Systemdateien unter "/home" noch in die **4. Banddatei** sichern:

  /home/oasys, /home/vmsys, /home/tele, /home/dsx

- den Rechner mit "init 6" neu hochfahren

  Die Sicherung der Benutzerdaten sollten Sie auf separaten Bändern durchführen.

Grundsystem sichern

Aufgabe: 7

a) Entwerfen Sie eine Prozedur, die die Dateisysteme "/", "/stand" und "/usr" mit "**cpio**" in eine große Banddatei sichert.

b) Entwerfen Sie eine Prozedur, die die Dateisysteme "/" und "/stand" mit "**cpio**" in eine kleine Banddatei sichert und "/usr" in die 2. Banddatei.

Lösungsvorschlag zu b):

Was passiert, wenn umountall nicht funktioniert und sich z.B. /home nicht abmontieren läßt?

Datensicherung

Aufgaben

Aufgabe 8:

Für "**mount**", "**umount**", "**chown**" stehen Ihnen spezielle Kommandos zur Verfügung, die Ihnen gezielt "Super-User"-Rechte verleihen. **Rufen Sie daher kein Kommando mit einem absoluten Pfadnamen auf.**

Erzeugen Sie sich wieder ein eigenes Dateisystem (mit "mkfs" wie in Aufgabe 3)und montieren Sie Ihr Dateisystem.

a) Kopieren Sie mit "**cpio**"-Kommando alle Dateien und Dateiverzeichnisse unterhalb von "/home/sv0" in das Verzeichnis, an dem Ihr Dateisystem montiert ist (nicht in Ihr HOME-Dateiverzeichnis !).

b) Sichern Sie nun Ihr komplettes Dateisystem mit dem "**cpio**"-Kommando in eine Plattendatei "$HOME/cpio_dat".

c) Sichern Sie Ihr Dateisystem zusätzlich mit dem "**tar**" in die Datei "$HOME/tar_dat".

d) Löschen Sie die Dateien in Ihrem montierten Dateisystem (nicht die in Ihrem HOME-Verzeichnis !).
Restaurieren Sie Ihre Daten mit "**cpio**" aus der Datei "$HOME/cpio_dat".

Zusatzaufgabe:

e) Montieren Sie Ihr Dateisystem ab. **Wenn Sie sicher sind, daß Ihr Dateisystem nicht mehr montiert ist,** löschen Sie es mit "**mkfs**".

Montieren Sie das leere Dateisystem erneut, ändern Sie den Eigentümer und restaurieren Sie Ihre Daten mit "**cpio**" aus Ihrer Sicherungsdatei.

f) Der Kursleiter legt eine 3 1/2-Zoll-Diskette in das Laufwerk. Versuchen Sie, Daten von dieser Diskette einzulesen. Welches Format hat die Diskette?

Aufgabe 9:

Löschen Sie einige Dateien in Ihrem Dateisystem.

Restaurieren Sie **nur** die gelöschten Dateien aus Ihrer Sicherungsdatei "cpio_dat" und anschließend noch einmal mit "**tar**" aus der "tar_dat".

Aufgabe 10 (Streamer):

Bitte probieren Sie Ihre Lösung erst aus, wenn Sie vom Kursleiter die Magnetbandkassette erhalten und eingelegt haben.

Schreiben Sie folgende Prozedur:

- Spulen Sie das Band zurück, vermeiden Sie Nachspannen

- `echo "Hallo SINIX" `date` >/dev/tapen` # 0. Banddatei

- Sichern Sie alle Dateien Ihres HOME-Dateiverzeichnisses in der nächsten Banddatei, ohne das Band zurückzuspulen. (1. Banddatei).

- Sichern Sie den Inhalt von "/home/sv0" mit relativen Pfadnamen in der nächsten Banddatei:

- Spulen Sie das Band zurück

- Lesen Sie die Banddateien 0 und 2 in einem Unterdateiverzeichnis von Ihrem HOME-DVZ ein.

Sicherung ganzer Dateisysteme

Das "ufsdump"-Kommando:

- Absolut- oder Differenzsicherungen ganzer **"ufs"-Dateisysteme** (z.B. "/home", "/home1", ...).
- sichert auch Gerätedateien und läßt Zugriffszeiten und Zugriffsrechte unverändert (wie eine physikalische Sicherung)
  —> selbst das Datum des letzten Lesezugriffs bleibt unverändert
- sichert auch bei der Absolutsicherung nur die tatsächlich belegten Blöcke des Dateisystems
- erlaubt verschiedene Stufen von Differenzsicherungen ("Dump-Level": 0 bis 9)
- **sollte nur auf abmontierte und überprüfte Dateisysteme angewendet werden** (sinnvollerweise im Einbenutzer-Betrieb)

  —> andernfalls ist die Sicherung eventuell nicht wieder einlesbar
- ist auch zur Datensicherung auf Platte verwendbar
- "ufsdump"-Sicherungen können von "ufsrestore" wieder eingelesen werden

Die Sicherungsstufen von ufsdump

Stufe 0: Absolutsicherung eines Dateisystems

Stufe n: "ufsdump" sichert alle Dateien des Dateisystems, deren Modifikations-Datum neuer ist als das Datum der letzten notierten Sicherung mit einer Sicherungsstufe < n.

Wird das "ufsdump"-Kommando mit dem Schalter "-u" aufgerufen, hinterlegt es folgende Informationen in der Datei **/etc/dumpdates :**

- Plattenbereich, der gesichert wurde
- Datum und Sicherungsstufe der Sicherung

ufsdump / ufsrestore

Die Kommandos "ufsdump" und "ufsrestore" funktionieren im wesentlichen wie "dump" und "restore" in SINIX V5.2x oder in anderen Berkeley-Systemen.

Unterschiede zu SINIX V5.2x:

- die Kommandos stehen im "usr"-Dateisystem oder benötigen die "shared libraries" des "usr"-Dateisystems.

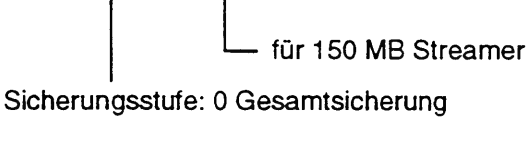
Sie sind daher nicht geeignet, das "usr"-System zu restaurieren !

- vor dem 1. Aufruf von "ufsdump" müssen Sie eventuell die Datei **"/etc/dumpdates"** einrichten:

```
>> /etc/dumpdates
```

Beispiel für "ufsdump" auf /home:

```
# als "root" anmelden !!!
cd /
SLICE=c0d0s4  # "slice" fuer /home (siehe /etc/vfstab)

# In Einbenutzer-Betrieb wechseln, alle Prozesse
# beenden, Dateisysteme abhängen:

init 1              # in Einbenutzer-Betrieb herunterfahren
sleep 180
killall 15          # Prozesse geordnet beenden
sleep 10
killall 9           # restliche Prozesse brutal beenden
sleep 5
umountall           # alles abmontieren (auch /home)
mount /usr          # "ufsdump" steht im "/usr" oder  benötigt
                    # "/usr/lib/libc.so.?"
fsck -y /dev/rdsk/$SLICE || exit 1
>> /etc/dumpdates
ufsdump -0usf 9000 /dev/tape /dev/rdsk/$SLICE
```



Beispiel für "ufsrestore" auf /home:

Bevor Sie mit "ufsrestore -r" ein Dateisystem einlesen können, müssen Sie mit "mkfs" ein neues leeres Dateisystem erzeugen. Dazu müssen Sie die "length" des betreffenden "slices" kennen:

```
# als "root" anmelden !!!
prtvtoc /dev/rdsk/c0d0s0    # liefert "length"

# von "Hand" die Größe von slice 4 lesen
...
LENGTH=102000        # in diesem Beispiel: Größe des slice 4

cd /
DSYS=/home
SLICE=c0d0s4         # "slice" fuer /home (siehe /etc/vfstab)

# In Einbenutzer-Betrieb wechseln, alle Prozesse
# beenden, Dateisysteme abhängen:

init 1               # in Einbenutzer-Betrieb herunterfahren
sleep 180
killall 15           # Prozesse geordnet beenden
sleep 10
killall 9            # restliche Prozesse brutal beenden
sleep 5
umountall            # alles abmontieren (auch /home)
mount /usr           # "ufsrestore" bentötigt "/usr"

# Dateisystem neu erzeugen und überprüfen:
mkfs -F ufs /dev/rdsk/$SLICE $LENGTH 34 15 || exit 1
#                              Werte für MC1558 !!!
fsck -F ufs -y /dev/rdsk/$SLICE || exit 1

mount $DSYS          # frisches Dateisystem montieren
                     # (muß in /etc/vfstab stehen)
cd $DSYS
ufsrestore -rf /dev/tape
```

Interaktives "ufsrestore"-Kommando

```
cd ????      # zuerst in das richtige Montierverzeichnis
             # wechseln
ufsrestore ivf /dev/tape
```

* liest die Indexeinträge der auf dem Band mit "ufsdump" gesicherten Dateien ("Inhaltsverzeichnis")

* stellt eine Shell-ähnliche interaktive Schnittstelle zur Verfügung (Prompt "restore> ")

* Sie können sich den Dateibaum auf dem Band ansehen und die Dateien auswählen, die "restore" einlesen soll

Mögliche Kommandos :

- ls
- cd
- pwd

- add [DATEI] a [DATEI]	DATEI (oder Dateiverzeichnis) zum späteren Einlesen auswählen —> in Extraktionsliste eintragen
- delete [DATEI] d [DATEI]	DATEI (oder DVZ) wieder aus der Extraktionsliste löschen

- help oder ?
- quit

- **extract**	**alle Dateien der Extraktionsliste von dem Band oder den Bändern einlesen**

Beispiel für ein Band:

```
restore> extract
specify next volume #: 1
set owner/mode for '.' [yn]: y
```

Hinweis zu "ls" :

"ls" kennzeichnet alle Dateien, die bereits in der Extraktionsliste stehen, mit einem "*".

Spiegelplatten (VPSS)

Lit.: SINIX V5.40/Virtuelles Platten-Subsystem
Benutzerhandbuch, U6768-J-Z145-1

Ziel:

Bei einem Fehler oder Ausfall der Hardware (z.B. Plattenfehler) sollen:

- keine Datenverluste auftreten
- das Computer-System sollte in möglichst kurzer Zeit wieder einsetzbar sein

Lösungsansatz:

Alle Daten müssen auf mehreren Platten mit eigenen Platten-Controllern gleichzeitig abgespeichert werden.

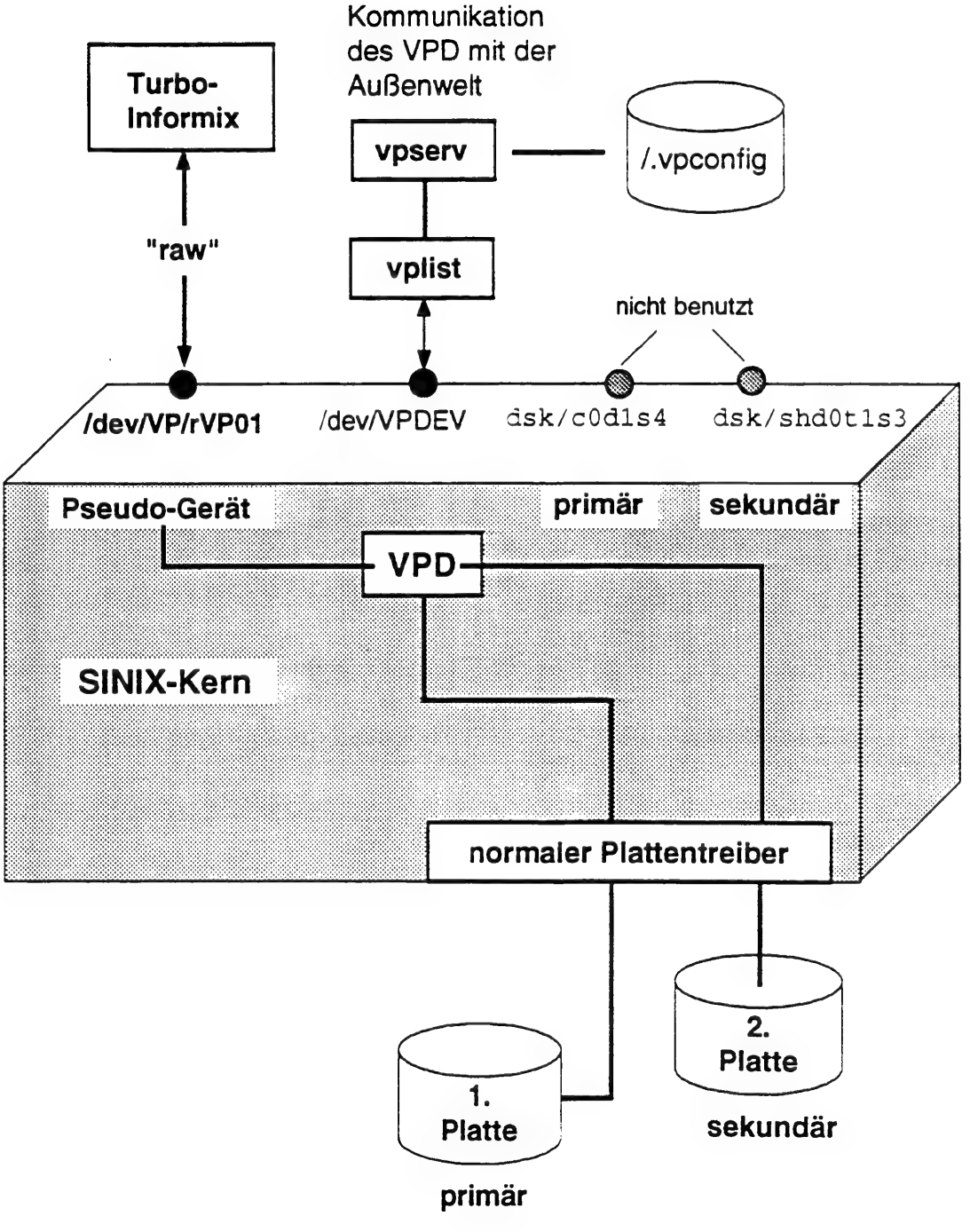
- Bei System- und Anwendersoftware und bei Konfigurations-Dateien, die sich selten ändern (z.B. "root"- und "usr"-Bereich), können Sie die betreffenden Dateien oder Plattenbereiche nach jeder Änderung auf eine Sicherungsplatte kopieren. Zu diesem Zweck benötigen Sie kein Spiegelplatten-System.

- Bei Datenbanken, die laufend verändert werden, sollte **jede Veränderung gleichzeitig auf mehreren Platten erfolgen.**

  Das **"Virtuelle Platten-Subsystem"** (VPSS) ermöglicht diese Datenvervielfachung für Anwendungen, die **direkt mit "raw"-Gerätedateien** von Plattenbereichen arbeiten (z.B. Turbo-Informix, Informix-ONLINE, Oracle oder Ingres).
  Für Plattenbereiche, die ein Dateisystem enthalten, ist VPSS erst ab SINIX V5.41 einsetzbar !

  Anstatt mit einer normalen "raw"-Gerätedatei müssen Sie die Anwendung mit einem **Pseudogerät** (virtuelle Platte) verbinden. Der VPD (VP-Treiber) im SINIX-Kern leitet alle Daten, die das Pseudogerät erhält, gleichzeitig an einen primären und eventuell mehrere sekundäre Plattenbereiche (Kopien) weiter.

  Die Zuordnung zwischen Pseudogeräten, primären und sekundären Plattenbereichen muß in einer binären Datei "/.vpconfig" hinterlegt sein.


Kommunikation
des VPD mit der
Außenwelt
Turbo-
Informix
vpserv
/.vpconfig
"raw"
vplist
nicht benutzt
/dev/VP/rVP01
/dev/VPDEV
dsk/c0d1s4
dsk/shd0t1s3
Pseudo-Gerät
primär
sekundär
VPD
SINIX-Kern
normaler Plattentreiber
2.
Platte
1.
Platte
sekundär
primär

Dateiverzeichnisse und Dateien für VPSS:

`/sbin/VP/..`

Dateiverzeichnis mit allen Kommandos und Dienstprogrammen des VPSS

`/dev/VP/..`

Dateiverzeichnis für Pseudo-Geräte:

rVP01 erste Pseudogerätedatei
rVPmax letzte Pseudogerätedatei

Weitere Pseudo-Gerätedateien müssen Sie mit dem "mknod"-Kommando anlegen. Sie müssen den Typ "c" und die gleiche Majornummer angeben wie bei "rVP01". Die Minornummer muß zwischen den Minornummern von "rVP01" und "rVPmax" liegen.

```
cd /dev/VP
ls -l rVP01    # Majornummer merken
/sbin/mknod rVP02 c MAJOR 2
```

`/dev/VPDEV`

Gerätedatei für "vpserv" und "vplist" zur Kommunikation mit dem VP-Treiber (VPD).

`/etc/inet/services`

Das VPSS bedient sich der gleichen Mittel zur Interprozeß-kommunikation wie LAN1 (TCP/IP). Für VPSS müssen zwei zusätzliche Zeilen in der Datei "/etc/inet/services" Ihres Netzverwaltungsrechners vorhanden sein:

```
vpserv_cmd 1230/tcp
vpserv_brd 1232/udp
```

Fehlen diese beiden Zeilen, so müssen Sie sie ergänzen und anschließend folgende Kommandos eingeben:

```
cd /var/yp
make services
```

Beispiel für eine ASCII-Konfigurationsdatei

Im folgenden Beispiel wird 1 Plattenbereich gespiegelt:

- Pseudogerät: /dev/VP/rVP01
  Primärer Bereich: /dev/dsk/c0d1s4
  Sekundärer Bereich: /dev/dsk/shd0t1s3 (Kopie)

Achtung: Sie müssen hier die "block"-Gerätedateien eintragen, obwohl VPSS auf die "raw"-Gerätedateien zugreift.

Tritt ein Schreibfehler auf, deaktiviert VPSS den fehlerhaften Bereich und schaltet ggf. auf einen anderen primären Bereich um.

Beispiel: ASCII-Datei VPCONFIG

```
# Pseudo-Geraet unter /dev/VP/..
rVP01 SHADOW
            /dev/dsk/c0d1s4 /dev/dsk/shd0t1s3
            PLEN=228960
#
```

Mit PLEN geben Sie die Länge des Plattenbereichs in Blöcken von 512 Bytes an.

VPSS starten/konfigurieren:

Im Verzeichnis "/sbin/VP" steht eine "rc"-Prozedur zur Verfügung, die als Beispiel dienen kann.

Erweitern Sie zunächst Ihre PATH-Variable:

```
PATH=/sbin/VP:$PATH
export PATH
```

Falls Sie eine Ereignisprotokollierung wünschen, können Sie zunächst das Programm "vpmsg" starten:

```
vpmsg /dev/console &            # Beispiel !
```

Anschließend starten Sie die Hintergrundprozesse, die Ihnen eine Kommunikation mit dem VP-Treiber ermöglichen:

```
vpserv -b /home1/.vpconfig &        # Beispiel !
```

Der "vpserv"-Prozeß startet automatisch den "vplist"-Prozeß. Die Datei "/home1/.vpconfig" dient in diesem Beispiel als Duplikat der binären Konfigurationsdatei "/.vpconfig".

Falls Sie noch nicht konfiguriert haben, erhalten Sie die Fehlermeldung, daß die Datei "/.vpconfig" fehlt. Die Fehlermeldung ist beim ersten Starten normal. Sind die Prozesse "vpserv" und "vplist" aktiv, so können Sie Ihre ASCII-Konfigurationsdatei in das binäre Format umsetzen und gleichzeitig in den Kern laden:

```
vpreq -c VPCONFIG # ASCII-Datei: VPCONFIG
```

Sie können sich den aktuellen Status des VPD ansehen mit dem Kommando:

```
vpreq -q
```

Achtung: "vpreq" funktioniert nur, wenn "vpserv" und "vplist" aktiv sind !

Virtuelle Partition nach Fehler wiederherstellen ("backup"):

```
vpreq -b rVP01
```

Key-Diskette

Bei neuen MX300 Rechnern ist das Betriebssystem bereits ab Werk auf der Platte vorinstalliert.

Sie müssen das Betriebssystem beim ersten Systemstart mit der mitgelieferten "KEY"-Diskette ("K.DISK", "AUTHORIZATION FILE") aktivieren.

Wenn Sie den Rechner zum ersten mal einschalten, erhalten Sie nach den normalen Meldungen beim Systemstart die Aufforderung:

```
Please install your SINIX authorization file
Insert diskette into Floppy Drive 1.
Type [go] when ready,
or [q] to quit:
```

Legen Sie Ihre Key-Diskette ein und geben Sie ein: **go**

Nach der Installation erhalten Sie die gleiche Aufforderung noch einmal:

```
Insert diskette into Floppy Drive 1.
Type [go] when ready,
or [q] to quit:
```

Geben Sie nun ein: **q**

Anschließend müssen Sie eventuell noch ein Korrekturband oder Korrekturdisketten installieren.
Den Umfang des vorinstallierten Betriebssystems entnehmen Sie Ihrer SINIX-Freigabemitteilung.

Vorinstalliertes Betriebssystem

MX300-Rechner werden normalerweise bereits im Werk auf der 1. Platte vorinstalliert.

Sie müssen "**nur**" noch (in dieser Reihenfolge):

- beim ersten Systemstart die Key-Diskette "**K.Disk**"("**SINIX authorization file**") installieren. Ohne Key-Diskette fährt das System nicht hoch !
- ev. ein aktuelles Korrekturband "**SINIXK**" installieren (Version prüfen: `uname -a`)
- Bildschirmarbeitsplätze und Drucker konfigurieren
- nicht benötigte Pakete deinstallieren: pkgrm ...
- eventuell die 2. Platte aktivieren
  Aus den Meldungen beim Systemstart erkennen Sie, welche Platten in Ihren Rechner eingebaut sind (Plattentyp !).
- eventuell den Rechnernamen ändern

  Der Rechner- und der Domain-Name ist beim vorinstallierten System auf "**SNI**" gesetzt.
- Kennwörter ändern:
  Die Benutzer "**root**", "**admin**", "**tele**", "**sysadm**" und "**install**" haben beim vorinstallierten System das Kennwort: "SNI"
- Anwendersoftware installieren
- Benutzergruppen und Benutzerkennungen einrichten

Rechnernamen ändern:

Es gibt kein Kommando "**hostname**" !

1) Rechner vom LAN abhängen ("**admin**"-Menü oder "**/etc/default/inet**" ändern und "**init 6**")

2) `sh /etc/init.d/inetinit down` # Prozesse beenden

3) Rechnernamen über "sysadm" ändern: "node name"
   /etc/net/ticots/hosts, ../ticlts/hosts, ../ticotsord/hosts
   Achtung: "System name" muß "SINIX-L" bzw. "SINIX-M" bleiben.

4) `uname -S RECHNERNAME` # Rechnername einstellen

5) `init 6` # Reboot

sysadm in V5.4

UNIX System V.4 bietet dem Systemverwalter ein fenster-orientiertes Menüsystem "sysadm" (OA&M - Menüs) in englischer Sprache. Die Menüdateien finden Sie unterhalb von "/usr/sadm/sysadm/menu".

Sie können das "sysadm"-Menü aufrufen:

- aus der "root"-Kennung mit dem **Kommando : "sysadm"**

- direkt über die spezielle **Benutzerkennung: "sysadm"**

"sysadm" bietet Ihnen folgendes Hauptmenü:

UNIX System V/386 Operations, Administration and Maintenance
1 UNIX System V Administration

applications	Administration for Available Applications
backup_service	Backup Scheduling, Setup and Control
buses	Bus Information and Management
diagnostics	Diagnosing System Errors
file_systems	File System Creation, Checking and Mounting
machine	Machine Configuration, Display and Shutdown
network_services	Network Services Administration
networks	Network Administration
ports	Port Access Services and Monitors
preSVR4	Peripherals Setup
restore_service	Restore From Backup Data
schedule_task	Schedule Automatic Task
software	Software Installation and Removal
storage_devices	Storage Device Operations and Definitions
system_setup	System Name, Date/Time and Initial Password Setup
users	User Login and Group Administration

In SINIX V5.4 sind mehrere Menüpunkte "abgeklemmt"(z.B. "users" oder "backup"), da Sie dafür das "admin"-Bediensystem (COLLAGE) oder Kommandos verwenden sollen.

Sie können "sysadm" bereits mit einer Menüauswahl als Argument aufrufen:

```
sysadm system_setup  #  Rechnernamen: "node name" ändern
sysadm networks
```

Installation von Softwareprodukten

Sie wollen zusätzliche SINIX-Softwareprodukte installieren, beispielsweise:

HIT, SIPLAN, INFORMIX, COLLAGE, OCIS, SIDRAW,
SICHART, ONLINE-Dokumentation, CCP-STA?, ...

Lesen Sie für **jedes Softwareprodukt** die produktspezifischen Installationshinweise im Benutzerhandbuch oder in der Freigabemitteilung des Softwareprodukts bevor Sie mit der Installation beginnen.

Softwareprodukte können Sie entweder über das "**admin**"-Bediensystem oder über das Kommando "pkgadd" installieren.
Die Installation, Deinstallation und Verwaltung von Softwareprodukten erfolgt in V5.4 über die Kommandos:

pkgadd Installation von SINIX/UNIX 5.4 Softwarepaketen

```
/usr/sbin/pkgadd -d ctape1          # vom Streamer

/usr/sbin/pkgadd -d diskette1       # 3,5 Zoll Diskette
```

Mit "**pkgadd**" können Sie nur Softwareprodukte installieren, die im "**package**"-Format vorliegen (neu in UNIX System V.4).
Softwarepakete für System V.3/386 müssen Sie mit dem "**custom**"-Kommando installieren.

pkginfo Informationen über installierte Softwareprodukte ausgeben:

```
pkginfo | pg                       # Übersicht über alle Pakete
pkginfo -l SIces                    # alle Informationen zu SIces
```

pkgrm Deinstallation von Softwarepaketen

```
/usr/sbin/pkgrm                     # Menüauswahl
```

Sie erhalten eine Übersicht der installierten Pakete und können menügeführt die Pakete auswählen, die Sie deinstallieren wollen.

```
/usr/sbin/pkgrm rfs lp  S1sim       # Pakete deinstallieren
```

/etc/device.tab

In der Datei /etc/device.tab sind Geräte und Geräteeingenschaften mit symbolischen "alias"-Namen ("ctape1", "diskette1", "diskette2", "disk1",...) verbunden. Die Datei dient als Informationsdatei und wird bei der Installation von Softwareprodukten verwendet. In der Datei sind unter anderem folgenden Geräteeigenschaften (Attribute) hinterlegt:

alias	symbolischer Name für das Gerät
bdevice	Pfadname der "block"-Gerätedatei (falls vorhanden)
bufsize	optimale Puffergröße zum Schreiben/Lesen
capacity	Kapazität in 512 Byte-Blöcken
cdevice	Pfadname der "raw"-Gerätedatei
desc	Beschreibung des Geräts (Kommentar)
erasecmd	Kommando zum Löschen der Daten
fmtcmd	Kommando zum Formatieren
norewind	Gerätedatei, bei der kein Rückspulen erfolgt
mkfscmd	Kommando, um Dateisystem zu erzeugen

Es sind nur Eigenschaften (Attribute) belegt, die für das Gerät relevant sind.

Kommandos:

Sie haben **Kommandos** zur Verfügung, mit denen Sie die Datei "/etc/device.tab" lesen oder bearbeiten können:

symbol. Gerätenamen ausgeben	:	`getdev`
Geräteattribute abfragen	:	`devattr [-v] alias attr`
Geräteattribute anlegen oder moifizieren	:	`putdev -m alias attr=...`
Geräteattribute löschen	:	`putdev -d alias attr`

Software-Pakete

SINIX V5.4 enthält neben dem Basissystem zahlreiche Softwarepakete.

Bei allen Rechnern, mit Ausnahme des MX300-45, wird empfohlen, alle Pakete zu installieren.
Sie können nachträglich einige Pakete wieder deinstallieren (z.B. "rfs" oder "lp").

Pakete, die SINIX-Erweiterungen gegenüber dem AT&T UNIX-System V.4 enthalten, haben einen Namen, der mit "SI" beginnt.

Kurzbeschreibung der Pakete bzw. Paketgruppen:

SIcoldrv SIcolrts SIcolface	COLLAGE-Laufzeitsystem und Bediensystem (muß installiert werden)
SIFonts SIMetapr	Druckprogramm für COLLAGE und COLLAGE- Anwendungen
SImshrs SImesui	Menü-Laufzeitsystem (MES) Menü-Bediensystem
	wird z.B. benötigt zur Administration von LAN
SIlpr	SINIX Spoolsystem V3.1
lp	Alternative zu SIlpr: das AT&T Spoolsystem (nicht installieren)
SIcompat	Kommandos aus SINIX V5.2x (muß installiert werden)
terminf ed	terminfo-Paket: Sie benötigen "siemens.ti" Editor Paket: "**vi**", "**spell**", ... muß beides installiert werden)

Pakete zur Netzanbindung (TCP/IP, LAN1):

nsu inet rpc	Basispakete zur Netzanbindung und zur Interprozeßkommunikation über "sockets"
SIexos201	SINIX Treiber für Ethernet Baugruppe
SIlanadm	SINIX LAN Administration (Menü)

Zusätzlich (bei Bedarf):

nfs dfs	Verteiltes Dateisystem (NFS bzw. DFS) "**Network File System**"
rfs	Alternative zu NFS: "**Remote File Sharing**"

Software-Pakete (Fortsetzung)

C-Entwicklungssystem (nur für Programmierer):
(fehlt beim MX300-45)

scde	Standard C Entwicklungssystem von AT&T
SIces	SINIX Erweiterungen zum CES
SICEScp, SIcp	Interaktives Programmiersystem auf Basis von COLLAGE: ColProg

Unterstützung von SINIX-Besonderheiten: (nicht MX300-45)

SIsim	Terminalanschluß über TACSI (SIM-Baugruppe)
SIscsi	SCSI-Unterstützung für Exabyte, Wormlaufwerke, externe SCSI-Platten,
...	
SIvpss	VPSS Spiegelplatten Unterstützung

Für das AT&T Menüsystem (z.B. "sysadm") benötigen Sie:

fmli	AT&T Form and Menu Language Interpreter
oam	Operations, Administration and Maintenance (es genügt, wenn Sie das "**BASIC user interface**" installieren)

Weitere Pakete:

xcp	XENIX-Kommandos, MS-DOS Disketten bearbeiten, XENIX-Gerätedateien für Disketten
compat	Berkeley-Kommandos und Bibliotheken (/usr/ucb/*)
termcap	termcap-Kompatibilität
face windowing	AT&T Framed Access Command Environment AT&T Fenstersystem

nicht notwendig

Die Installation der Pakete erfolgt in zufälliger Reihenfolge.

Sie können jedes Paket nachträglich installieren oder deinstallieren:

```
/usr/sbin/pkgadd -d ctape1        # von SINIX3 installieren
                                  # dauert sehr lange !
```

Erzeugen von Softwarepaketen: (für Entwickler)

2 Formate für "packages"

- Dateibaum-Struktur — auf Platte / s5-Dateisystem auf Floppy

- "datastream"-Format für beliebige Datenträger

"package" erzeugen:
pkgmk-Kommando: erzeugt Dateibaumstruktur

"package" umsetzen (z.B. in "datastream"):
pkgtrans-Kommando

Informationsdateien für pkgmk:

pkginfo-Datei (müssen Sie erstellen)
prototype-Datei: Liste des gesamten Pakets mit allen Informationen

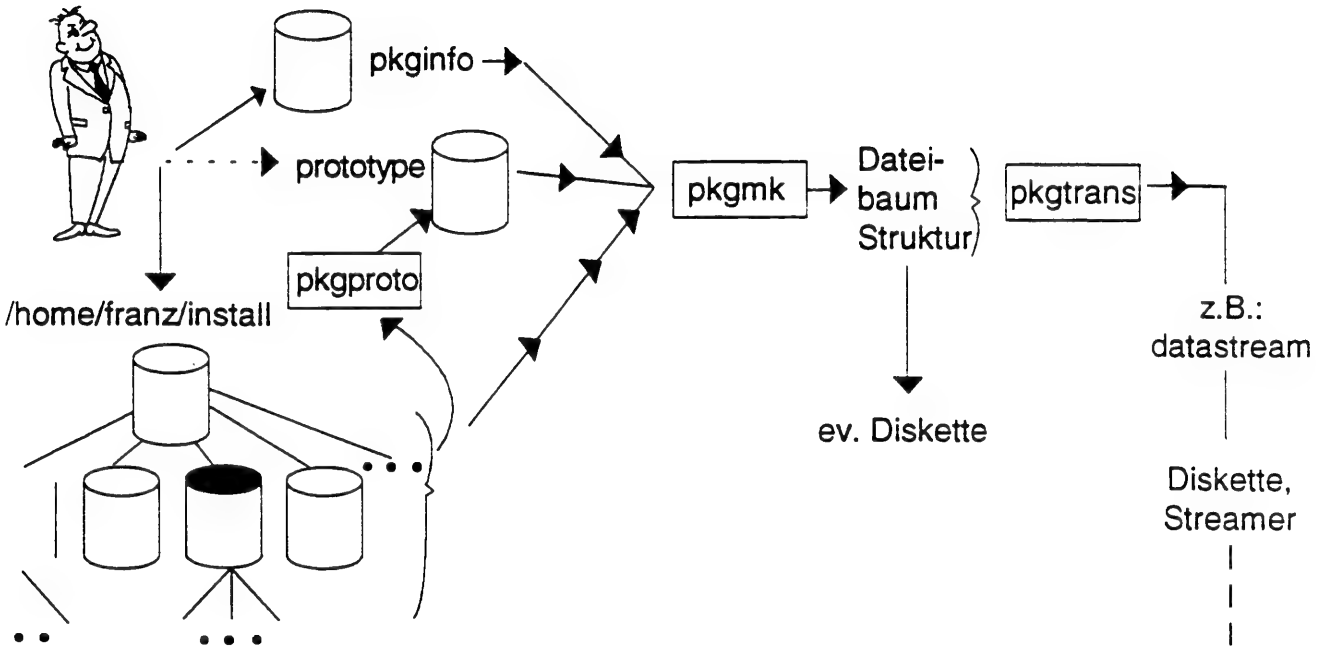
Hilfsprogramm zum Erzeugen der prototype-Datei:

```
echo "i  pgkinfo"  > prototype
pkgproto  /home/franz/install=/opt >>prototype
```

unterhalb dieses Pfadnamens muß sich ein genaues Abbild des Pakets befinden.

Paket auf Diskette erzeugen:

```
pkgmk -d /dev/rdsk/f03ht mon
```



Dateien, Verzeichnisse,
Programme aus denen
das Paket besteht
(mit richtigen Zugriffsrechten)

Datei: pkginfo

siehe pkginfo (4)

```
* PGK=mon                    Package-Name
* NAME=Software Monitor
  CLASSES=none               muß übereinstimmen mit "Class" in "prototype"
* VERSION=1
* CATEGORY=applications
  VENDOR=Walter Mecky
* ARCH=i386
  •
  •
  •
```

(*) Pflichtfelder

Literatur:

SINIX V5.4 Systemschnittstellen und Werkzeuge für die Anwendungs-programmierunng

Leitfaden für Programmierer: U6414-J-Z95-1

Kapitel: "Packaging von Anwendersoftware"

Installation auf dem MX300

Installationssatz

- SINIX0, SINIX1, SINIX2 Disketten
- SINIX3 Magnetbandkassette
- K.Disk Key-Diskette: "SINIX authorization file"
- **SINIX** K Korrekturband: **SI**update

In der Liefereinheit "**Message-Files**" ist zusätzlich enthalten:

- README Diskette Softwarepaket mit aktueller Freigabemitteilung

Installationsablauf (MX300)

⇨ SINIX0 einlegen und Rechner einschalten Aufforderung abwarten und:

⇨ SINIX1 einlegen und bestätigen Aufforderung abwarten und:

⇨ SINIX2 einlegen und bestätigen

⇨ Sie können wählen, ob Sie das System nur auf der 1. Platte oder gleich auf beiden Platten installieren wollen. Wenn Sie das System nur auf der 1. Platte installieren, können Sie die 2. Platte nachträglich aktivieren (und beliebig einteilen).

⇨ eine UNIX-Partition einrichten (100%, ACTIVE) (wie bei "**fdisk**")

⇨ Typ der Dateisysteme "**/**", "**/usr**", "**/home**", "**/var**" und "**/opt**" festlegen ("**ufs**" oder "**s5**").

optional: "**/tmp**" Dateisystem anlegen
Die Installationsprozedur bietet Ihnen nicht an, weitere "**slices**" einzurichten.

⇨ Größe der Dateisysteme in Zylindern bzw. in Sektoren festlegen (vgl. SINIX-Freigabemitteilung)

⇨ Die Installationsprozedur erzeugt die Dateisysteme auf der Festplatte und installiert ein Minisystem auf der Festplatte.

⇨ Das System "**bootet**" von der Festplatte und die weitere Installation erfolgt vom SINIX3-Band (siehe übernächste Seite)

Installation auf dem MX500-90

Installationssatz:

- SINIX0 Magnetbandkassette (155 MB-Streamer-Band)
- K.Disk Key-Diskette "SINIX authorization file"
- eventuell weitere Korrekturbänder oder Disketten

Installationsablauf (MX500):

- SINIX0-Band einlegen, Taste AUTOBOOT ausschalten, Rechner einschalten und auf das * des Hardwaremonitors warten. Falls der Rechner automatisch hochfährt, müssen Sie AUTOBOOT ausschalten und noch einmal beginnen.
- Bootkommando eingeben, um vom Band zu booten:

```
b0 ts(32,1) root=ramd(0) swap=ramd(1)
```

Das root-Dateisystem wird nun automatisch in die RAM-Disk eingelesen.

- AUTOBOOT wieder einschalten, damit die MX500 in Zukunft automatisch hochfahren kann.
- Die Installationsprozedur stellt fest, welche Platten angeschlossen sind. Sie können eine **"primary disk"** und eine **"secondary disk"** für die Installation auswählen.
- Sie werden gefragt, wie Sie "/stand", "swap", "/" "/usr", "/home", "/var", "/opt" und "/tmp" auf diese beiden Platten verteilen wollen. Sie können auch noch weitere Slices für Dateisysteme und für Arbeitsspeicherabzüge ("/dev/dump") konfigurieren.
- Die Installationsprozedur erzeugt die Dateisysteme auf der Festplatte und installiert ein Minisystem auf der Festplatte.
- Das System bootet von der Festplatte. Anschließend wird automatisch das SINIX-Grundsystem (Basissystem) eingespielt (siehe nächste Seite).

Installation vom SINIX3 bzw. SINIX0-Band

Die weitere Installation vom SINIX-Band besteht aus 3 Teilen:

1) Installation eines Basissystems (20 Minuten)

2) Eingabe von Kennwörtern und Netzwerknamen und der Internetadresse

3) Installation der verschiedenen Software-Pakete

⇨ MX300: bis zu 3 1/2 Stunden mit vielen Abfragen
⇨ MX500: ca. 1 Stunde mit wenigen Abfragen

Installationsanleitung:

Eine genaue Installationsanleitung finden Sie in der SINIX-Freigabemitteilung.

Platteneinteilung:

Schreiben Sie sich bei der Installation die Größe jedes "**slices**" und den Dateisystemtyp ("**s5**" bzw. "**ufs**") auf !
Die Größe eines Zylinders hängt vom Plattentyp ab.

Notreparatur

Angenommen, Ihre "**/**" oder "**/stand**" Dateisysteme sind so defekt, daß Ihr Rechner nicht mehr hochfährt.

In diesem Fall bleiben Ihnen nur die Boot-Disketten und Ihre (hoffentlich vorhandene) "**cpio**"-Sicherung von "**/**" und "**/stand**". Glücklicherweise befindet sich auf der "SINIX2"-Diskette das Kommando "**cpio**".

Falls Plattenfehler die Ursache für Ihr Problem sind, müssen Sie zunächst die Plattenfehler beseitigen lassen (TDS-Diskette) oder die Platte austauschen, bevor Sie die Dateisysteme restaurieren.

Booten von Diskette oder vom SINIX0-Band:

- Beginnen Sie die Installation, wie auf den Seiten L 5-10 bzw. L 5-11 beschrieben.
- Booten von SINIX0, für Minirootsystem und swap-Bereich werden RAM-Disks verwendet.
  Beim MX300 wird die SINIX2-Diskette unter /install an die RAM-Disk montiert.
- Brechen Sie die Installation bei der ersten echten Abfrage mit <DEL> ab.
- Sie befinden sich nun auf der RAM-Disk.
  Beim MX300 dürfen Sie die montierte SINIX2 nicht entnehmen.

Wir müssen 2 Fälle unterscheiden:

A) Ihre 1. Platte ist noch in Ordnung oder repariert, die Platteneinteilung ist unverändert, so daß nur die Dateisysteme überprüft (eventuell neu erzeugt) und die fehlenden Daten eingespielt werden müssen.

B) Ihre 1. Platte wurde ausgewechselt oder Sie wollen die Platteneinteilung verändern.
In diesem Fall rufen Sie auf:

```
fdisk /dev/rdsk/c0d0s0                  # UNIX-Partition einrichten

disksetup -IB /dev/rdsk/c0d0s0          # "slices" einrichten
                                        # Dateisysteme erzeugen
```

Dateisysteme prüfen (erzeugen) und Daten einspielen:

Auf dem RAM-Disk-System haben Sie nur wenig Kommandos zur Verfügung! Anstelle von "**ls**" können Sie "**echo ***" verwenden.

Falls "**/**"-Dateisystem auf der Platte vom Typ "**ufs**" ist:

```
# eventuell Dateisystem neu einrichten ?
# /etc/fs/ufs/mkfs /dev/rroot LENGTH NSECT NTRACK ?

/etc/fs/ufs/fsck -y /dev/rroot          # prüfen

/etc/fs/ufs/mount /dev/root /mnt        # montieren
```

Das Dateisystem "**/stand**" (mit Systemkern):

```
# eventuell Dateisystem neu einrichten ?
# /etc/fs/bfs/mkfs /dev/rdsk/c0d0s10 LENGTH ?

/etc/fs/bfs/fsck /dev/rdsk/c0d0s10      # prüfen

mkdir /mnt/stand                        # sollte bereits vorhanden sein !

/etc/fs/bfs/mount /dev/dsk/c0d0s10 /mnt/stand
```

Wie prüfen und montieren wir das "**/usr**"-Dateisystem ?

PATH erweitern, damit Sie Kommandos der Platte zur Verfügung haben:

```
PATH=$PATH:/mnt/usr/bin:/mnt/usr/sbin:
export PATH
```

"cpio"-Sicherungen einlesen:

Die "**cpio**"-Sicherungen müssen mit relativem Pfadnamen gesichert sein !!!!!

```
cd /mnt                          # "/"-Bereich der Platte
umask 0

cpio -ivBdmu < /dev/rmt/c0s0     # überschreiben
                                 # im Minisystem existiert kein "/dev/tape"
```

"**cpio**" überschreibt normalerweise keine Dateien neueren Datums sondern meldet solche Dateien als "**Fehler**". Wenn Sie den Schalter "**-u**" angeben, überschreibt "**cpio**" auch neuere Dateien.

```
cd /
init 6                           # neu hochfahren
```

Systemgrenzen, Systemparameter

Informationsdatei "mtune"

UNIX System V.4 besitzt ca. 170 Systemparameter, die in der Datei **"/etc/conf.d/cf.d/mtune"** aufgelistet sind.

Sie finden dort die Namen der Systemparameter, die Standardwerte, Minimal- und Maximalwerte und Kommentare.

In der Datei "mtune" sollten Sie keine Änderungen vornehmen!

Konfigurationsdatei "stune"

Wollen Sie einen Systemparameter ändern, so rufen Sie das Kommando auf:

```
/etc/conf/bin/idtune NAME  Wert
```

"idtune" überprüft den neuen Wert und trägt ihn in die Datei **"/etc/conf/cf.d/stune"** ein. In dieser Datei finden Sie alle Systemparameter, deren Wert vom Standardwert abweichen soll.

Habne Sie einen Wert geändert, müssen Sie den Systemkern neu binden. Dazu rufen Sie ein Kommando auf:

```
/etc/conf/bin/idbuild
init 6                    # System neu starten
```

Einige Systemgrenzen entfallen:

- Einige typische Systemgrenzen früherer UNIX-Systeme entfallen in SINIX V 5.4, da die betreffenden Systemtabellen jetzt dynamisch angelegt werden.

- Andere Systemgrenzen könne Sie in V5.4 prozeßspezifisch über den C-System-aufruf **"setrlimit ( )"** setzen und mit **"getrlimit ( )"** abfragen.

- Eine genaue Beschreibung der Systemparameter finden Sie im Handbuch: "Leitfaden für Systemverwalter".

Einige Systemparameter

Einige wichtige Systemparameter können Sie der folgenden Liste entnehmen:

MAXUP — maximale Anzahl Prozesse pro Benutzer (zum Schutz des Systems)
sinnvolle Werte: 20 - 400

NPROC — maximale Anzahl Prozesse im System

NCALL — sinnvoller Wert: >= NPROC

ARG_MAX — maximale Größe der Argumentlisten beim Starten von Programmen
sinnvolle Werte: 5120 - 20480

SEGMAPSZ — **Anzahl der 8 KB Blöcke, die maximal für den virtuellen "Disk-Cache" herangezogen werden.** Dies ist einer der wichtigsten Tuning-Parameter! Ist SEGMAPSZ zu klein, kann der Systemkern die Plattenzugriffe nicht optimal puffern.
Ist SEGMAPSZ zu groß, bleibt zu wenig Speicher für die laufenden Prozesse.
sinnvolle Werte: 1024 bei 16 MB Speicher
2048 bei 32 MB Speicher

NINODE — maximale Anzahl von "s5"-Inodes im Systemkern
Falls Sie ausschließlich "ufs"-Dateisysteme verwenden, kann dieser Wert klein sein.

UFSINODE — maximale Anzahl von "ufs"-Inodes im Systemkern
sinnvoller Wert: 1000

Besonderheiten des MX500

Bedienung des Hardware-Monitors

Der Hardwaremonitor meldet sich mit einem "*" an der ausgewählten Konsole (entweder C oder S).

kd0	deutschen Zeichensatz für lokale Konsole einstellen
kd1	deutschen Zeichensatz für Service-Konsole einstellen
ki0	internationalen Zeichensatz für lokale Konsole (Standardeinstellung)

wd=jjmmddhhmm.ss	Hardware-Uhr stellen (Greenwich Meantime)
rd	Hardware-Uhr lesen ("read date")
ra	Bootflags lesen
rl	die letzten Konsol-Meldungen lesen (z.B. panic-String)

Gerätenamen und Nummern im Monitor

Der Hardwaremonitor spricht Geräte nicht über ihre SINIX-Gerätedateinamen an, sondern verwendet Namen in einer betriebsystemunabhängigen- **(bsu-)-Darstellung:**

```
gt(gn,le)[dateiname] [parameter]
```

- le — Nummer der logischen Einheit (Banddateinummer / Slice-Nummer, beginnt bei 0)
- gn — Gerätenummer nach Formel
- gt — Gerätetyp: ts, sp, is oder sd

ts — Magnetband-Kassetten-Laufwerk am SCSI-Bus
sp — neue ("ssd" bzw. SPA)-Platten am synchronen SCSI-Bus (MX500-90)
is — ESDI-Platten am Multibus (MX500-75)

Typischer Bootstring für MX500-90:

b 0 sp(0,10)unix root=ssd(1) swap=ssd(2) rootfstype=ufs

Besonderheiten des MX500

Gerätenamen und Nummern

Formel für Platten am synchronen SCSI-Bus (MX500-90):

gn = Kanal * 64 + Anschluß * 8

Formel für Platten am Multibus (MX500-75):

gn = Multibus * 512 + Controller * 8 + Anschluß

bsu- Darstellung	SINIX- **Treiber**	SINIX- **Gerätedatei**	Bedeutung
ts(32,4)	ct	c0s0 /dev/tape	Banddatei Nr.4 (Streamer)
sp(0, 10)	ssd(10)	c0d0s10 /dev/tape	Slice 10 0. Kanal, 0.Anschluß
sp(8, 1)	ssd(129)	c0d1s1	Slice 1 0. Kanal, 1.Anschluß
sp(64,1)	ssd(1025)	c1d0s1	Slice 1 1. Kanal, 0.Anschluß
is(0,10)	hd(10)	c0d0s10	Slice 10/0.Platte
is(1,1)	hd(1025)	c0d1s1	Slice 1/1.Platte

siehe Formel: 0*64+1*8 / 1*64+0*8

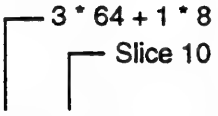
Minornummer

Besonderheiten des MX500

bsu-Programme booten

MX500-90:

Booten von einer Platte am 3. Kanal, 2. Anschluß:



b 0 sp(200,10)unix root=ssd(3201) swap=ssd(3202) rootfstype=ufs

ssd(Minornummer)

= Kanal * 1024 + Anschluß * 128 + Slice

Booten vom Installationsband:

b 0 ts(32,1) root=ramd(0) swap=ramd(1)

Banddatei Nr. 1

Weitere Kommandos:

wn0=sp(0,10)unix rootfs=ufs	boot-Argument in NVRAM schreiben
wn1=sp(0,0) sp(0,6)	dump-Argument in NVRAM schreiben
b 80	dump (Arbeitsspeicherabzug) erzeugen

Weitere Besonderheiten des MX500

Kommandos: online und offline

Mit den Kommandos "online" und "offline" können Sie Prozessoren aktivieren und deaktivieren:

```
online  -a

offline 1
```

Kommando: /sbin/bootflags

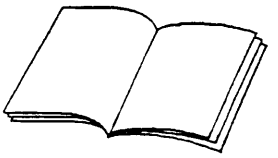
Mit diesem Kommando können Sie die bootflags im laufenden System aus dem NVRAM lesen oder in das NVRAM hineinschreiben:

```
bootflags -p -b 'sp(0,10)unix root=ssd(1) swap=ssd(1) swap=ssd(2) rootfs=ufs
```

Alternativer Bootsring für Arbeitspeicherabzug:

```
bootflags -p  -a 'b80 sp(0,0) sp(0,6) 0 /dev/dump'
```

Anhang



Literaturhinweise

- Pflichtlektüre für den SINIX-Systemverwalter sind die SINIX-Freigabemitteilung und die Manuale der Angebotseinheiten:

  "Messagefile" : SINIX-SPR-D (für deutsch)

  Freigabemitteilung gedruckt und auf Diskette mit
  Manualkorrekturen
  Bediensystem für Systemverwalter

  Dokumentationspaket: SINIX-DOC-D (für deutsch)

  Kommandos Teil 1 bis 3
  COLLAGE-Bediensystem
  SINIX SPOOL V3.0 / V3.1
  Referenzhandbuch für den Systemverwalter
  Leitfaden für den Systemverwalter

  Das Handbuch "Leitfaden für den Systemverwalter" beschreibt UNIX System V.4 . SINIX V5.4 weicht jedoch in einigen Punkten von dieser Dokumentation ab (z.B. in der Terminalverwaltung).

  Die beiden Angebotseinheiten müssen Sie gesondert bestellen

  Zum Umstieg von SINIX V5.2x auf SINIX V5.4 kann Ihnen der "Portierungs-leitfaden" (Band 1) gute Dienste leisten.
  Bestellnummer: U 6313-J-Z99-1

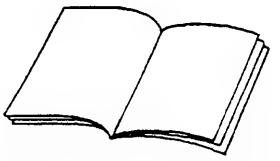
Weitere Literatur für Systemverwalter basiert meist auf UNIX System V.3 oder auf Berkeley- bzw. Sun-Systemen.Sie benötigen daher einige Kenntnisse, um die Informationen herauszufiltern, die auch für SINIX V5.4 gültig sind.

Beispiele:

- UNIX Systemverwaltung/ Hauke Richter / Addison-Wesley (Deutschland) 1991/ ISBN 3-89319-297-2 (Beschreibung von UNIX System V.3)

- UNIX System Administration Handbook/ E. Nemeth, G. Snyder, Scott Seebass/ Prentice-Hall 1989/ ISBN 0-13-933441-6

Anhang



Nachschlagewerke für den Systemverwalter:

- SINIX V5.4 Kommandos: A - K und L - Z
  für Benutzer und Systemverwalter

- Referenzhandbuch für Systemverwalter
  ein Handbuch mit mehreren Kapiteln:

  - Kommandos für Systemverwalter z.B. fsck (1M)

    (Kapitel 1 der klassischen UNIX-Gliederung)

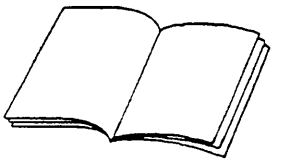
  - Dateiformate z.B. passwd (4)

    (Kapitel 4 der UNIX-Gliederung)

  - Include Dateien z.B. vtoc.h (5)

  - Geräte z.B. fd (7)

Vorsicht: In der ersten Ausgabe dieser deutschen Handbücher haben sich einige Übersetzungsfehler eingeschlichen.

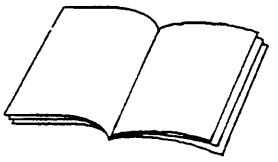


cleanup

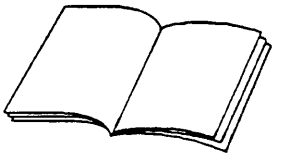
```
#! /sbin/sh
#  private cleanup-Prozedur fuer SINIX V5.4
#  1 mal in der Woche aufrufen: Freitag 12:00
#            z.B.: mit "crontab"-Kommando !!!
#
#  am Freitag Nachmittag sollte der Rechner
#  dann heruntergefahren werden
#
# PATH darf nicht mit Doppelpunkt beginnen !!!!
PATH=/sbin:/usr/bin:/usr/sbin:/opt/bin:/opt/etc:/usr/ucb
export PATH
#
# Aufraeumarbeiten unter /var/adm
cd /var/adm
#
cp klog.msg Oklog.msg      # letzte Version aufbewahren
> klog.msg                 # Datei auf 0 setzen
#
mv sulog Osulog
> sulog                    # Datei auf 0 setzen
#
cp wtmp Owtmp              # letzte Version
cp wtmpx Owtmpx            # letzte Version
> wtmp
> wtmpx
#
# Aufraeumarbeiten: /var/cron/log und /var/crash
#
cp /var/cron/log /var/cron/Olog
> /var/cron/log
#
rm -f /var/crash/*
#
# Aufraeumarbeiten: /tmp und /var/tmp
#
cd /tmp
find . -type f -mtime +3 -exec rm {} \;
cd /var/tmp
find . -type f -mtime +3 -exec rm {} \;
#
# alte core-Dateien im gesamten System loeschen
# (find-Aufruf dauert sehr lange)
cd /
find / -name core -atime +7 -exec rm {} \;
```

save_vtoc Platteneinteilung sichern

```
#! /bin/sh
# save_vtoc: V1.0 13.11.91
#####################################################################
# Prozedur zur Sicherung der Platteneinteilungen in Dateien,
# die als Eingabe fuer "disksetup -d" verwendbar sind.
#
# Siemens Nixdorf Training Center
#      D9 DS TC 21
#
# Ausgabedateien fuer jede Platte:
#
#      VTOC.c?d? Ausgabedateien von prtvtoc -f
#      DISK.c?d? (verwendbar fuer disksetup -d)
#      DISK.0, DISK.1 (Verweise auf DISK.c?d?)
#####################################################################
# temporaere Dateien loeschen
trap '/bin/rm [a-z]*.c[01]d[0-1]' 0 2 3
#
# Feststellen, welche Platten vorhanden sind, VTOC.c?d? erzeugen:
cd /tmp
DISKS=
for d in c0d0 c0d1 c1d0 c1d1
do
    if prtvtoc -f VTOC.$d /dev/rdsk/${d}s0 2>/dev/null
    then
      DISKS="$DISKS $d"
      # Tabulatoren entfernen
      tr -s '\11' ' ' < VTOC.$d > vtoc.$d
    else
      rm VTOC.$d
    fi
done
#
# fuer jede Platte Standardwertdatei DISK.c?d? erstellen
#
i=0
for d in $DISKS
do
  # Ueberschrift, Slice 0, Slices mit Groesse 0 und
  # Slice 7 aus vtoc.c?d? entfernen:
ed - vtoc.$d <<\stop
1,2d
1,$s/^ //
g/ 0$/d
/^7 /d
w
q
stop
  #
```

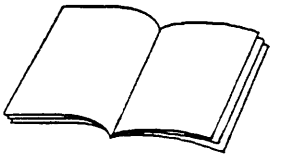


```
  # In /etc/vfstab nachsehen, welcher slice wo montiert ist
  # unterschiedliche Behandlung wegen root-slice auf 1. Platte
  case $d in
  c0d0) tr -s '\11' ' ' < /etc/vfstab |
          sed -n "/rroot/{s:/dev/root:1:
                          s: / : /root :p
                          }
                  /$d/s:/dev/dsk/${d}s::p" > vfstab.$d ;;
  *) sed -n "/$d/s:/dev/dsk/${d}s::p" /etc/vfstab > vfstab.$d ;;
  esac
  #
  # std_datei zusammenbauen und Zeilen aus vtoc-Datei loeschen
  > std_datei.$d
  while read SLICE RAW DVZ DSTYP REST
  do
    # Slice-groesse aus vtoc.$d heraussuchen
    set `grep "^$SLICE " vtoc.$d` 0 0 0 0 0
    echo "$SLICE $DVZ $DSTYP $5" >> std_datei.$d
    # Zeile in vtoc.$d loeschen
    grep -v "^$SLICE " vtoc.$d > tmp.$d
    mv tmp.$d vtoc.$d
  done < vfstab.$d
  #
  # Alle Bereiche hinzufuegen, die nicht in der /etc/vfstab
  # sondern nur in der vtoc-Datei stehen
  while read SLICE TAG FLAGS START SIZE
  do
    case $TAG in
    *3) echo "$SLICE swap - $SIZE" >> std_datei.$d ;;
    *) echo "$SLICE r$SLICE - $SIZE" >> std_datei.$d ;;
    esac
  done < vtoc.$d
  #
  # Auf jeder Platte einen Slice variabel mit "100 W" angeben,
  # um kleine Unterschiede in den Plattengroessen auszugleichen
  #
  egrep -v '/root|/usr|/stand|/opt|/tmp|swap' std_datei.$d > nonsystem.$d
  #
  ZAHL=`cat nonsystem.$d | wc -l`
  if [ "$ZAHL" -gt 0 ]
  then MAX=`awk '$4 > max { name=$2; max=$4 }
                 END { print name }' nonsystem.$d`
  else MAX=/xxxx
  fi
  # megabytes ausrechnen
  awk '$2 != MAX { MB = $4 / 2048
       printf"%s\t%s\t%s\t%3.0fM\n",$1,$2,$3,MB
       }
       $2 == MAX {
       printf"%s\t%s\t%s\t100W\n",$1,$2,$3
       }' MAX=$MAX std_datei.$d | sort +0n -1 | > DISK.$d
  ln DISK.$d DISK.$i
  i=`expr $i + 1`
done
```

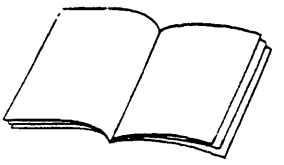


system_sich Systemsicherung

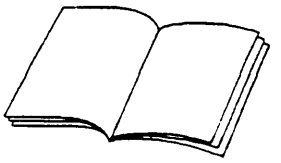
```
#! /sbin/sh
# system_sich V2.0 18.11.91
#################################################################
# Prozedur zur Systemsicherung in Verbindung mit "save_vtoc"
#                  und einer eigenen Prozedur "cleanup"
#
# sichert:    Platteneinteilung (siehe "save_vtoc")
#             und Label                                  0. Banddatei
#             "/" und "/stand"                           1. Banddatei
#             "/usr" und "/var"                          2. Banddatei
#             "/opt"                                     3. Banddatei
#             /home/oasys,/home/vmsys,/home/tele         4. Banddatei
#
# Der Rest von /home muss getrennt gesichert werden (z.B. mit DSX)
# Falls "/opt" zu gross ist, muss anstelle der 3. und 4. Banddatei
# ein neues Band verwendet werden !!!
#################################################################
PATH=/sbin:/usr/bin:/usr/sbin:/usr/ucb:/opt/bin
export PATH
exec > /dev/sysmsg 2>&1      # Ausgabe auf Konsole umlenken
# Ist Band eingelegt ?
sync
mt -f /dev/tape noret || { echo "Kein Band eingelegt"; exit 1 ; }
mt -f /dev/tape ret & # Straffen im Hintergrund
#
cd /
sync
init 1              # in run level 1 herunterfahren
sleep 70
killall 15          # alle Prozesse sanft beenden
sleep 5
killall 9           # restliche Prozesse brutal beenden
sleep 5
#
wait                # auf mt-Kommando warten
#
echo "Dateisysteme abhaengen"
umountall           # fast alles abmontieren
umount /proc        # wichtig !!
umount /dev/fd >/dev/null 2>&1    # eventuell unnoetig
#
# Dateisysteme wieder montieren:
mount /stand >/dev/null 2>&1      # nur zur Sicherheit
mount /usr     # "find" und "cpio" stehen im /usr-Bereich
mount /var
mount /opt
mount /tmp     # fuer Protokoll der Sicherung
#
```



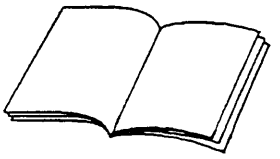
```
#
exec > /tmp/Protokoll 2>&1   # Protokolldatei
cat /dev/klog &              # Kernelmeldungen mitlesen
PID=$!
tail -f /tmp/Protokoll >/dev/sysmsg 2>&1 & # Protokoll auf Konsole ausgeben
PID="$PID $!"
#
echo "Sicherungsprotokoll `date`"
#
# aufraeumen:
echo "/var/crash, /var/adm, /var/cron und /tmp bereinigen"
cleanup
#
# Platzbedarf berechnen, Platteneinteilung feststellen,
# Label erzeugen, alles in 0. Banddatei sichern
#
cd /tmp
echo    "Systemsicherung vom `date` im cpio -oc Format
        Platteneinteilung und Label    0. Banddatei
        "/" und "/stand"               1. Banddatei
        "/usr" und "/var"              2. Banddatei" > Label
#
# Platzbedarf berechnen
#
MB1=`df -k | egrep "root|stand|usr|var|opt" |
          awk '{ summe = summe + $3 }
               END { printf"%3.0f", summe/1024 }'`
MB1=`echo $MB1`
MB2=`df -k | egrep "root|stand|usr|var" |
          awk '{ summe = summe + $3 }
               END { printf"%3.0f", summe/1024 }'`
MB2=`echo $MB2`
if [ "$MB1" -gt 135 ]
then
  echo "Zu viele Daten: $MB1 (einschl. /opt)"
  echo "/opt passt nicht auf das Band"
  echo "Sicherungsumfang: $MB2 MB"
  OPT=N
else
  echo "      /opt           3. Banddatei" >> Label
  echo "Sicherungsumfang: $MB1 MB"
  OPT=J
fi
if [ "$MB2" -gt 135 ]
then
  echo "Zu viele Daten: $MB2 (ohne /opt)"
  df -k
  echo "***** Abbruch *******"
  exit 1
fi
```



```
if [ "$MB1" -lt 125 ]
then
  echo " /home/oasys,vmsys,tele 4. Banddatei" >> Label
  HOMESYS=J
else
  HOMESYS=N
fi
cat Label
#
echo "Platteneinteilung sichern:"
rm -f DISK.* VTOC.*
#
save_vtoc || exit 1
#
ls Label DISK.* VTOC.* | cpio -ocBO /dev/tapen
#
# /, /stand in 1. Banddatei sichern:
#
echo "Sicherung von / und /stand :"
#
# Liste aller Dateien im / Dateisystem ohne /usr/*,
#                                  /var/*, /tmp/* erzeugen:
cd /
for d in `ls -a`
do
  case $d in
  .) echo . ;;     # wegen der Zugriffsrechte
  ..) : ;;
  usr|var|opt|tmp) echo $d ;; # nur das leere Verzeichnis
  home*) echo $d ;;
  *) if test -d $d          # bei Verzeichnissen
    then find $d -print     # "find" aufrufen
    else echo $d            # nur Dateinamen ausgeben
    fi ;;
  esac
done | cpio -ocBO /dev/tapen
#
# /usr und /var sichern:
echo "Sicherung von /usr und /var:"
find usr var -print |cpio -ocBO /dev/tapen
#
#
if [ "$OPT" = "J" ]
then
#    /opt sichern ********************************
  echo "/opt : "
  find opt -print |cpio -ocBO /dev/tapen
fi
```



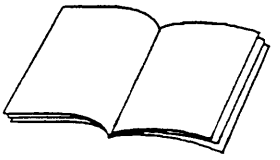
```
if [ "$HOMESYS" = "J" ]
then
#    /home/vmsys,oasys,tele sichern ***************************
  echo "/home/vmsys,oasys,tele:"
  mount /home
  find home/vmsys home/oasys home/tele -print | cpio -ocBO /dev/tapen
fi
# zurueckspulen
< /dev/tape
echo "Ende der Sicherung `date`"
sleep 2
kill $PID
mail root < /tmp/Protokoll
cp /tmp/Protokoll /Protokoll.$$
sync
sleep 10
init 6   # reboot
```



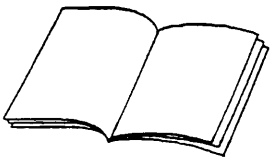
Abhängigkeiten der Systempakete von SINIX V5.4
(siehe auch Handbuch für MX500: Installation)

Paket-nr.	Paketname	abhängig von Paketnummer	Paketbeschreibung
1	SICEScp	4,9	Compiler-Support für Programmierumgebung auf COLLAGE-Basis
2	SIFonts	-	SINIX Metapr Font-Information
3	SIMetapr	2	SINIX Metapr
4	Slces	30	SINIX CES (C-Entwicklungs-System) SINIX CES (*c development system*)
5	Slcoldrv	-	COLLAGE Pseudo-Treiber und Line-Disziplin *COLLAGE pseudo driver and line discipline*
6	Slcolface	5,7,8	Benutzer-Schnittstelle auf SINIX-COLLAGE-Basis *SINIX COLLAGE based user interface*
7	Slcolrts	5,23,26	COLLAGE-Laufzeit-System *COLLAGE runtime system*
8	Slcompat	-	SINIX-Kompatibilität-Paket *SINIX compatibility package*
9	Slcp	7,18	Programmierumgebung auf SINIX-COLLAGE-Basis *SINIX COLLAGE based programmers environment*
10	Slexos	26	SINIX EXOS201 Ethernet LLC-Treiber *SINIX EXOS201 Ethernet LLC driver*
11	Sllanadm	6,13,14,23 26,29	SINIX-LAN-Administration *SINIX LAN administration*
12	Sllpr	-	SINIX-Drucker-Spooler *SINIX printer spooler*

Anhang



Paket-nr.	Paketname	abhängig von Paketnummer	Paketbeschreibung
13	Slmesui	14	Benutzer-Schnittstelle auf SINIX-MES-Basis *SINIX MES based user interface*
14	Slmshrs	-	SINIX-msh-Laufzeit-System *SINIX msh runtime system*
15	Slsxsi	-	SINIX-scsi-Geräte-Support *SINIX scsi device support*
16	Slsim	-	SINIX SIM-Board-Support *SINIX SIM board support*
17	Slvpss	-	Virtuelles SINIX-Platten-Subsystem *SINIX virtuell partition subsystem*
18	compat	26	Kompatibilitäts-Paket (ucb-Umgebung) *Compatibility package (ucb environment)*
19	dfs	-	DFS-Utilities *DFS utilities*
20	ed	-	Editier-Paket *Editing package*
21	face	20,32	AT&T-Umgebung für *framed* Zugriffs-Kommando *AT&T framed access command environment*
22	fmli	-	AT&T Form- und Menü-Sprachen-Interpreter *AT&T form and menu language interpreter*
23	inet	26	internet-Utilities *internet utilities*
24	lp	-	LP Druck-Service (AT&T) *LP print service (AT&T)*
25	nfs	26,29	Utilities des Netzwerk-Datei-Systems *Networking file system utilities*
26	nsu	-	Utilities des Netzwerk-Support *Networking support utilities*
27	oam	22	Betrieb, Administration und Wartung *Operation, administration and maintenance*
28	rfs	26	Utilities für Remote-Datei-Systeme *Remote file system utilities*



Paket-nr.	Paketname	abhängig von Paketnummer	Paketbeschreibung
29	rpc	26	Utilities für Remote-Prozedur-Aufruf *Remote procedure call utilities*
30	scde	-	Standard-C-Entwicklungs-Umgebung *Standard C development environment*
31	termcap	-	AT&T-Termcap-Kompatibilitäts-Paket *AT&T termcap compatibility package*
32	terminf	-	Utilities für Terminal-Information *Terminal information utilities*
33	windowing	-	Utilities für AT&T Fenstertechnik *AT&T windowing utilities*
34	xcp	-	XENIX-Kompatibilitäts-Paket *XENIX compatibility package*

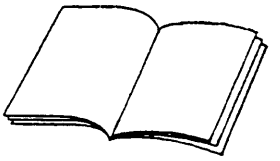
Auf einer MX300-45 dürfen Sie die folgenden Pakete **nicht** installieren, da diese für dieses System nicht freigegeben sind bzw. nicht unterstüzt werden:

Paketnummer	Paketname
1	SICEScp
4	Slces
9	Slcp
15	Slscsi
16	Slsim
17	Slvpss
21	face
28	rfs
30	scde

Folgende Pakete sollten Sie auf jeden Fall installieren, damit Sie ein arbeitsfähiges Basissystem zur Verfügung haben:

Paketnummer	Paketname
	Grundsystem
26	nsu
23	inet
8	Slcompat
5	Slcoldrv
7	Slcolrts
6	Slcolface
12	Sllpr
20	ed
18	compat

Anhang



Stichwortverzeichnis

Dateien und Verzeichnisse

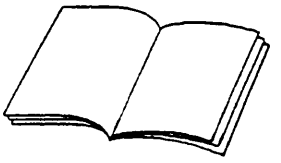
Seite

/default/boot ..... L 3 - 2
/dev/dump ..... L 2 - 10, L 3 - 9
/dev/rfl2 ..... L 4 - 13
/dev/rfl3 ..... L 4 - 13
/etc/default/ ..... L 1 - 20
/etc/default/inet ..... L 5 - 2
/etc/default/tar ..... L 4 - 16
/etc/device.tab ..... L 5 - 5
/etc/dfs/dfstab ..... L 1 - 27
/etc/group ..... L 1 - 17
/etc/inittab ..... L 3 - 13
/etc/mnttab ..... L 1 - 25
/etc/partitions ..... L 2 - 9
/etc/passwd, ... L 1 - 16
/etc/shadow ..... L 1 - 16
/etc/TIMEZONE ..... L 3 - 28
/etc/vfstab ..... L 1 - 22, L 1 - 27
"/opt"-Dateisystem ..... L 2 - 5
/stand/boot ..... L 3 - 2
"/usr"-Dateisystem ..... L 2 - 6
/usr/lib/libc.so.1 ..... L 2 - 7
"/var"-Dateisystem ..... L 2 - 5
/var/adm ..... L 3 - 23
/var/adm/klog.msg ..... L 3 - 3
/var/crash ..... L 3 - 9
/var/cron ..... L 3 - 23

A

Accounting ..... L 2 - 5
accounting ..... L 3 - 24
acctcom ..... L 3 - 24
admin ..... L 1 - 10, L 1 - 11
Alpha-COLLAGE ..... L 1 - 14
Alpha-COLLAGE Sondertasten ..... L 1 - 15
Anwendersoftware ..... L 2 - 5
Arbeitsspeicher ..... L 1 - 41
Arbeitsspeicherabzüge ..... L 3 - 9
at ..... L 3 - 26
AT&T Spoolsystem ..... L 5 - 6
AUTO-BOOT ..... L 3 - 7
AUTO-ON ..... L 3 - 7
AUTOBOOT ..... L 3 - 2
autopush ..... L 3 - 15

Anhang



Seite

B

Banddatei ..... L 4 - 6
Banddateien ..... L 4 - 5
Bandendeerkennung ..... L 4 - 3
Bänder kopieren ..... L 4 - 8
bcheckrc ..... L 3 - 18
Bedienfeld MX300 ..... L 3 - 6
Bedienfeld MX500 ..... L 3 - 7
Benutzergruppe ..... L 1 - 9
Benutzerkennung ..... L 1 - 9
Benutzerverwaltung ..... L 1 - 11
Besonderheiten des MX500 ..... L 1 - 4
bfs ..... L 2 - 19
"block"-Geräte ..... L 2 - 11
"block"-Gerätedatei ..... L 2 - 16
Blockungsfaktor ..... L 4 - 3, L 4 - 15
Boot ..... L 3 - 1
Boot-Dateisystem ..... L 2 - 4
boot-Parameter ..... L 3 - 2
Booten vom Installationsband ..... L 1 - 3
Booten von Diskette ..... L 5 - 14
bootflags ..... L 1 - 4
Bootstring für MX500-90 ..... L 1 - 1
bsu-Darstellung ..... L 1 - 1

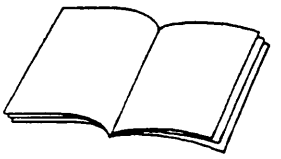
C

Cache ..... L 1 - 1
cleanup ..... L 3 - 23
cpio ..... L 4 - 4, L 4 - 9, L 4 - 10
“cpio”-Sicherungen einlesen ..... L 5 - 16
CPU-Benutzung ..... L 1 - 41
cron.allow ..... L 3 - 26
cron.deny ..... L 3 - 26
crontab ..... L 3 - 25
ctape1 ..... L 5 - 5

D

date ..... L 3 - 30
Dateibaum ..... L 2 - 3
Dateibaum kopieren ..... L 4 - 10, L 4 - 11
Dateibäume ..... L 1 - 21
Dateibäume sichern ..... L 4 - 10
Dateisystem ..... L 2 - 18

Anhang



Seite

Dateisysteme ..... L 1 - 21
Dateisystemtypen ..... L 2 - 19
Datenaustausch mit SINIX V5.2x ..... L 4 - 13
dd ..... L 4 - 18
dev ..... L 2 - 11
devattr ..... L 5 - 5
df ..... L 1 - 23
DIAG ..... L 3 - 6
diskadd ..... L 1 - 34
diskette1 ..... L 5 - 5
Disketten ..... L 4 - 12
disksetup ..... L 1 - 34, L 1 - 35
Drucker ..... L 2 - 11
dynamische “shared libraries” ..... L 2 - 7

E

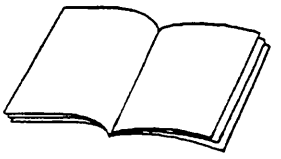
edvtoc -f ..... L 1 - 36
Einbenutzer-Betrieb ..... L 3 - 11
Einbenutzerbetrieb ..... L 3 - 20
EOF ..... L 4 - 5
EOF-Marken ..... L 4 - 5
Erzeugen von Softwarepaketen ..... L 5 - 8
Exabyte ..... L 4 - 1
externe SCSI-Platten ..... L 2 - 11

F

fdisk ..... L 1 - 34
Fehler im Dateisystem ..... L 3 - 19
Fehlerfall/Notfall ..... L 3 - 8
Fehlermeldungen ..... L 3 - 3
fsck ..... L 1 - 23, L 3 - 18, L 3 - 19, L 3 - 21
fsf off ..... L 3 - 18
fuser ..... L 1 - 25

G

Gerätedateien für Disketten ..... L 4 - 12
Gerätedateien für MB-Kassetten ..... L 4 - 2
getdev ..... L 5 - 55
groupadd ..... L 1 - 18
groupdel ..... L 1 - 18



Seite

H

Hardware-Monitors ..... L 1 - 1
Hardwarefehler ..... L 3 - 19
Hardwareuhr ..... L 3 - 28
Hardwareuhr stellen ..... L 3 - 30
herunterfahren ..... L 3 - 16

I

idtune 17
Indexnummer 2 ..... L 2 - 18
Informix-ONLINE ..... L 3 - 17, L 4 - 28
Informix-Online ..... L 2 - 10
init ..... L 3 - 11
inittab verändern ..... L 3 - 14
inodes ..... L 1 - 30
Installation auf dem MX300 ..... L 5 - 11
Installation auf dem MX500-90 ..... L 5 - 12
Installation von Softwareprodukten ..... L 5 - 4
Installationsprotokolle ..... L 2 - 5
iopb ..... L 3 - 5

K

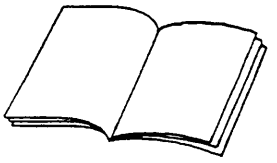
K.DISK ..... L 5 -1
Kanal ..... L 1 - 6
Kennwörter ..... L 1 - 16
Key-Diskette ..... L 5 - 1
killall ..... L 3 - 11, L 3 - 20
Kommandos für Dateisysteme ..... L 1 - 23

L

ldd ..... L 2 - 7
lost+found ..... L 3 - 22

M

Magnetband straffen ..... L 4 - 6
Magnetbandkassettenlaufwerke (Streamer) ..... L 4 - 1
Mehrbenutzer-Betrieb ..... L 3 - 11
Meldungen beim Systemstart ..... L 3 - 3
MET ..... L 3 - 28
MEZ ..... L 3 - 28
mkfs ..... L 1 - 23, L 1 - 28, L 1 - 29
mount ..... L 1 - 23, L 1 - 25, L 1 - 27



Seite

"mt": Bandkontrolle ..... L 4 - 6
Multiprozessor ..... L 1 - 3
MX 500 Modell 90 ..... L 1 - 4
MX500 ..... L 1 - 4
MX500 Modell 75 ..... L 1 - 4
MX500-90 ..... L 1 - 3

N

Neue Gruppe ..... L 1 - 12
neue Platte ..... L 1 - 34
Neuer Benutzer ..... L 1 - 12
NFS ..... L 1 - 27
Notfall ..... L 3 - 8
Notreparatur ..... L 5 - 14
nsect ..... L 2 - 13, L 1 - 29
ntrack ..... L 2 - 13, L 1 - 29

O

offline ..... L 1 - 4
online ..... L 1 - 4

P

"package"-Format ..... L 5 - 4
paging ..... L 2 - 10, L 1 - 41
Pakete auswählen ..... L 5 - 4
passwd ..... L 1 - 19
"passwd"-Kommando ..... L 1 - 19
physikalisch kopieren ..... L 4 - 18
pkgadd ..... L 5 - 4
pkginfo ..... L 5 - 4
pkgmk ..... L 5 - 8
pkgrm ..... L 5 - 4
pkgtrans ..... L 5 - 8
Platte konfigurieren ..... L 1 - 34
Platten-Aktivität ..... L 1 - 43
Plattenaktivitäten ..... L 1 - 41
Plattenbereich ..... L 2 - 9
Plattendaten ..... L 2 - 13
Platteneinteilung ..... L 2 - 9, L 1 - 22
Platteneinteilung nachträglich ändern ..... L 1 - 36
Postkörbe ..... L 2 - 5
Primärer Bereich ..... L 4 - 31

Seite

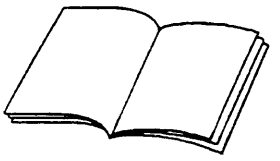
primary disk ..... L 5 - 12
Protokolldateien ..... L 2 - 5, L 3 - 23
Prozeduren automatisch starten ..... L 3 - 25
Prozeduren in /etc/rc?.d: ..... L 3 - 17
Prozeduren zum Starten von Prozessen ..... L 3 - 17
Prozessorleistung ..... L 1 - 1
prtvtoc ..... L 1 - 36
prtvtoc -f ..... L 2 - 9
prtvtoc -p ..... L 2 - 13
Pseudo-Dateisystem ..... L 2 - 4

R

RAM-Disk ..... L 5 - 12, L 5 - 14
"raw"-Geräte ..... L 2 - 11
"raw"-Gerätedatei ..... L 2 - 16
rc-Dateien ..... L 3 - 17
Rechnernamen ändern ..... L 5 - 2
RESET ..... L 3 - 6
root ..... L 1 - 10
run-level ..... L 3 - 11
russ ..... L 1 - 44

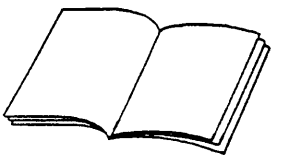
S

s5 ..... L 2 - 19
sar: system activity reporter ..... L 1 - 40
savecore ..... L 3 - 9
SCSI-Controller ..... L 4 - 1
SCSI-Geräte ..... L 1 - 8
SCSI-Kanäle ..... L 1 - 3
SCSI-Systembusadapter ..... L 1 - 6
secondary disk ..... L 5 - 12
Sektoren ..... L 2 - 13
Sekundärer Bereich ..... L 4 - 31
Shared Libraries ..... L 2 - 7
shutdown ..... L 3 - 16
Sicherung des Betriebssystems ..... L 4 - 19
Sicherung ganzer Dateisysteme ..... L 4 - 24
Sicherungen einlesen ..... L 4 - 10
Sicherungskommandos ..... L 4 - 9
Sicherungskonzept ..... L 4 - 19
SIM-Baugruppe ..... L 1 - 7
SINIX 5.4 ..... L 2 - 1
SINIX Spoolsystem V3.1 ..... L 5 - 6
SINIX-ODT ..... L 2 - 1



Seite

SINIX0 ..... L 5 - 11
SINIX1 ..... L 5 - 11
SINIX2 ..... L 5 - 11
SINIX3 Magnetbandkassette ..... L 5 - 11
slice ..... L 2 - 9
slices ..... L 1 - 36
Software-Pakete ..... L 5 - 6
Sommerzeit ..... L 3 - 29
Spiegelplatten ..... L 4 - 28
Stammbaum ..... L 2 - 1
STREAMS ..... L 3 - 15
Stromausfal ..... L 3 - 9
Stromausfall ..... L 3 - 19
stune ..... L 5 - 17
swap -a ..... L 1 - 38
“swap”-Bereich ..... L 1 - 38
swap-Bereiche ..... L 2 - 10
swapping ..... L 1 - 41
symbolische Links ..... L 2 - 8
symbolische Verweise ..... L 2 - 8
sysadm ..... L 1 - 10
sysadm in V5.4 ..... L 5 - 3
System name ..... L 5 - 2
System V.3 ..... L 2 - 1
System V.4 ..... L 2 - 1
System-Ressourcen ..... L 2 - 15
Systemabsturz ..... L 3 - 9
Systemaktivitäten messen ..... L 1 - 40
Systemaufrufe protokollieren ..... L 1 - 44
Systemauslastung ..... L 3 - 24
Systemgrenzen ..... L 5 - 17
Systemparameter ..... L 5 - 17, L 5 - 18
Systemsicherung ..... L 4 - 20
Systemstart ..... L 3 - 1, L 3 - 13
SystemV.3.2 ..... L 2 - 1
Systemverwaltung ..... L 1 - 11
Systemzeit ..... L 3 - 28



Seite

T
TACSI ..... L 1 - 7, L 2 - 11
TAK-Geräte ..... L 1 - 7
Taktfrequenz ..... L 1 - 1
tar ..... L 4 - 4, L 4 - 9, L 4 - 14, L 4 - 15, L 4 - 17
Terminals ..... L 2 - 11
Tuning-Parameter ..... L 5 - 18
TZ-Variable ..... L 3 - 28, L 3 - 29

U
ufs ..... L 2 - 20
ufsdump ..... L 4 - 4, L 4 - 9, L 4 - 24, L 4 - 25
ufsrestore ..... L 4 - 26
umount ..... L 1 - 23, L 1 - 25
umountall ..... L 1 - 25, L 3 - 20
uname ..... L 5 - 2
useradd ..... L 1 - 18
userdel ..... L 1 - 18

V
Verteilte Dateisysteme ..... L 1 - 27
Virtuelles Platten-Subsystem ..... L 4 - 28
vpmsg ..... L 4 - 32
vpreq ..... L 4 - 32
vpserv ..... L 4 - 32
VPSS ..... L 4 - 28, L 4 - 31
VTOC ..... L 2 - 9

W
Weltzeit ..... L 3 - 28
who -r ..... L 3 - 11
Winterzeit ..... L 3 - 29
Worm-Laufwerk ..... L 2 - 11

X
X/OPEN-Standard ..... L 2 - 2
XPG 2 ..... L 2 - 2
XPG 3 ..... L 2 - 2

Z
Zylinder ..... L 2 - 13

Inhalt

	Seite
Grundlagen und einfache Systemverwaltung	**L 1 - 1**
Rechnerleistung	L 1 - 1
Prozessorleistung in MIPS	L 1 - 2
Aktuelle SINIX-Rechner und Versionen	L 1 - 3
UNIX/SINIX-Stammbaum	L 1 - 5
X/OPEN-Standards	L 1 - 6
Korn Shell: ksh	L 1 - 7
Benutzer und Benutzergruppen	L 1 - 8
Menügeführte Systemverwaltung	L 1 - 10
Neuen Benutzer eintragen	L 1 - 12
Wichtige Tasten von Alpha-COLLAGE	L 1 - 13
Alpha-COLLAGE Sondertasten	L 1 - 14
/etc/passwd, /etc/shadow	L 1 - 15
/etc/group	L 1 - 16
Kommandos zur Benutzerverwaltung	L 1 - 17
Das "passwd"-Kommando	L 1 - 18
/etc/default/login, /etc/default/su	L 1 - 19
Weitere Kommandos zur Systemverwaltung	L 1 - 20
Kommandos, Tagesmeldung, Eröffnungsbilschirm	L 1 - 21
t-Bit bei "/tmp" und "/var/tmp"	L 1 - 22
Dateibaum in SINIX V5.4/UNIX V.4	L 1 - 25
Dateibaum in SINIX V5.4	L 1 - 26
Das "/var"-Dateisystem	L 1 - 27
Das "usr"-Dateisystem	L 1 - 28
Shared Libraries in V 5.4	L 1 - 29
Einfache symbolische Verweise	L 1 - 30
Hardware, Plattenbereiche und Dateisysteme	**L 2 - 1**
TACSI	L 2 - 3
Logische Platteneinteilung	L 2 - 4
Das Verzeichnis /dev	L 2 - 6
Gerätedateien und Plattenbereiche	L 2 - 7
"block" und "raw"-Gerätedateien	L 2 - 8
Was ist ein Dateisystem?	L 2 - 10
Plattentypen und Plattendaten	L 2 - 11
Dateisystemtypen	L 2 - 14
Dateisystemtyp "ufs"	L 2 - 15
Platteneinteilung: /etc/vfstab	L 2 - 17
Kommandos für Dateisysteme	L 2 - 18
fsck, mount, umount	L 2 - 19
mount, umount	L 2 - 20
Dateisysteme neu erzeugen?	L 2 - 22

Inhalt

	Seite
mkfs-kommando	L 2 - 23
Anzahl der "inodes"	L 2 - 24
"mkfs"-Parameter bei "s5"	L 2 - 24
Zusätzliche Platte konfigurieren	L 2 - 28
disksetup	L 2 - 29
Platteneinteilung nachträglich ändern	L 2 - 30
"swap"-Bereiche konfigurieren	L 2 - 32
Sinnvolle Plattenaufteilung	L 2 - 33
Verteilte Dateisysteme (NFS)	L 2 - 34
Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben	**L 3 - 1**
Bot von Diskette oder Platte	L 3 - 1
boot-Parameter	L 3 - 2
Meldungen beim Systemstart	L 3 - 3
Feststellen von Hardwarefehlern	L 3 - 4
Arbeitsspeicherabzüge	L 3 - 6
"run-level", init	L 3 - 7
run level 0, 1, 2 und 3	L 3 - 8
/etc/inittab	L 3 - 9
inittab verändern	L 3 - 10
Betriebssystem herunterfahren	L 3 - 11
Bedienfeld MX300	L 3 - 12
Bedienfeld MX500	L 3 - 13
rc-Dateien in V5.4	L 3 - 15
Fehler im Dateisystem	L 3 - 17
Einbenutzerbetrieb: run-level 1	L 3 - 18
"fsck" selbst aufrufen?	L 3 - 19
lost + found	L 3 - 20
Regelmäßige Kontrollaufgaben	L 3 - 21
Mehrfach-erfolglose Logins protokollieren	L 3 - 22
Prozeduren automatisch starten	L 3 - 23
Das "at"-Kommando	L 3 - 24
Datensicherung	**L 4 - 1**
Geräte zur Datensicherung	L 4 - 1
Arbeiten mit Magnetbandkassette	L 4 - 2
Pufferung	L 4 - 3
Grundprinzip von tar, cpio und ufsdump	L 4 - 4
Erzeugen mehrerer Banddateien	L 4 - 5
"mt" : Bandkontrolle	L 4 - 6
Bänder kopieren?	L 4 - 8
Sicherungskommandos	L 4 - 9

cpio	L 4 - 10
Dateibaum kopieren	L 4 - 11
Gerätedateien für Disketten in V5.4	L 4 - 12
Datenaustausch mit SINIX V5.2x	L 4 - 13
Das tar-Kommando	L 4 - 14
tar-Aufruf	L 4 - 15
tar	L 4 - 17
dd	L 4 - 18
Sicherungskonzept	L 4 - 19
Arbeitsschritte zur Systemsicherung	L 4 - 20
Grundsystem sichern	L 4 - 21
Sicherung ganzer Dateisysteme	L 4 - 25
ufsdump/ufsrestore	L 4 - 26
Interaktives "ufsrestore"-Kommando	L 4 - 28
Installation, Notreparatur, Konfiguration, Spoolsystem	**L 5 - 1**
Key-Diskette	L 5 - 1
Vorinstallieretes Betriebssystem	L 5 - 2
sysadm in V5.4	L 5 - 3
Installation von Softwareprodukten	L 5 - 4
Software-Pakete	L 5 - 5
Erzeugen von Softwarepaketen: (für Entwickler)	L 5 - 7
Datei pkginfo	L 5 - 9
Installation auf dem MX300	L 5 - 10
Installation auf dem MX500-90	L 5 - 11
Konfigurierung der Peripherie	L 5 - 13
Auswahlmöglichkeiten	L 5 - 14
Systemdateien	L 5 - 15
gettydefs und ttydefs	L 5 - 16
terminfo statt termcap	L 5 - 17
Eigenschaften von "terminfo"	L 5 - 18
Die SINIX Druckerverwaltung	L 5 - 19
Die CONFIG-Datei	L 5 - 20
Neustart des Spoolsystems mit neuer CONFIG	L 5 - 21
Notreparatur	L 5 - 22
sar: system activity reporter	L 5 - 25
Beispiele für sar	L 5 - 27
Auswertung der Messdateien	L 5 - 28
truss: Systemaufrufe protokollieren	L 5 - 29
Systemgrenzen, Systemparameter	L 5 - 30

Besonderheiten des MX500 **L 6 - 1**
Bedienung des Hardware-Monitors L 6 - 1
Gerätenamen und Nummern L 6 - 2
bsu-Programme booten L 6 - 3
MX500 - Die wichtigsten Daten im Vergleich L 6 - 4

Lösungen **1 - 10**

Grundlagen und einfache Systemverwaltung

Rechnerleistung

Die Leistung eines Mehrplatzrechners hängt ab von folgenden Leistungsmerkmalen :

- **Prozessorleistung** (wird oft in "MIPS" angegeben)
- **Arbeitsspeicher**
- **Festplatten** und **Plattensteuerungen** (Controller)
- Schnittstellen zur Peripherie (z.B. E/A-Prozessoren)
- Interne Datenübertragung (Bussysteme)

Prozessorleistung:

CPU: Central Processing Unit - Prozessor

MMU: Memory-Management Unit - Adreßübersetzung / Arbeitsspeicherverwaltung

FPU: Floating-Point-Unit - Prozessor für Gleitpunktrechnung

Bei modernen Prozessoren (NSC 32532, INTEL 80486)
sind CPU, MMU und FPU in einem Baustein ("Chip") zusammengefaßt.

Taktfrequenz:

Je höher die Taktfrequenz, desto höher ist theoretisch die Rechenleistung. Bei Taktfrequenzen ab 25 MHz muß der Prozessor jedoch häufig auf Befehle oder Daten warten, die erst aus dem Arbeitspeicher geholt werden müssen.

Moderne Prozessoren besitzen daher:

Prozessor-Cache: Schnelle Zwischenspeicher für Befehle und Daten zum Vermindern der Arbeitsspeicherzugriffe

Second Level Cache: Zusätzlicher schneller Zwischenspeicher auf der CPU-Baugruppe

Pipeline: Der Prozessor liest Befehle bereits im voraus und holt sich die zugehörigen Daten

Prozessorleistung in MIPS

Die Prozessorleistung von Rechnern mit ähnlicher Prozessorarchitektur kann in "MIPS" angegeben werden und untereinander verglichen werden. Sie müssen dabei 3 verschiedene "MIPS"-Werte unterscheiden:

- Angaben des Prozessorherstellers ohne Betriebssystem

  Beispiel: **20 MIPS** für INTEL 80486 mit 25 MHz

- "MIPS"-Werte, die unter fast realen Bedingungen mit optimierten Tests gemessen wurden

  Beispiel: **15 MIPS** für MX300-50 mit 80486/25 MHz

- "MIPS"-Werte für kommerzielle Last, die unter den gleichen Bedingungen gemessen wurden, unter denen der Kunde mit dem Rechner arbeitet

  Beispiel: **9 MIPS** für MX300-50 mit 80486/25 MHz (kommerzielle Leistung)

Falls Sie "SINIX MIPS" mit "BS2000 MIPS" vergleichen wollen, müssen Sie die kommerzielle Leistung der SINIX-Rechner mindestens durch 2.5 dividieren.

Die "MIPS"-Angabe macht keine Aussage darüber, ob ein Rechner für Ihre Anwendungen geeignet ist !

Aktuelle SINIX-Rechner und Versionen

SINIX V5.2x

MX300-15/-30:

Prozessor: NSC 32532, Taktfrequenz 25 MHz (max. 16 MB Speicher)
kommerzielle Leistung: 5.4 MIPS

MX500-75: Multiprozessor

Prozessoren: NSC 32532, Taktfrequenz 25 MHz (max. 64 MB Speicher)
bis zu 8 Prozessoren0
Platten: interne ESDI-Platten am Multibus I (1,6 MB/s)

kommerzielle Leistung: max. 42 MIPS (5.25 MIPS pro Prozessor)

SINIX V5.4

MX300-45/50:

Prozessor: INTEL 80486, Taktfrequenz 25 MHz
kommerzielle Leistung: 9.0 MIPS

MX300-60:

Prozessor: INTEL 80486, Taktfrequenz 33 MHz, 4 Mbit-Speicherchips
kommerzielle Leistung: 12.0 MIPS

MX300-60/3:

wie MX300-60 aber mit 128 KB Second Level Cache

MX300-60/5:

Prozessor: INTEL 80486, Taktfrequenz 50 MHz,
256 KB Second Level Cache

MX500-90: Multiprozessor

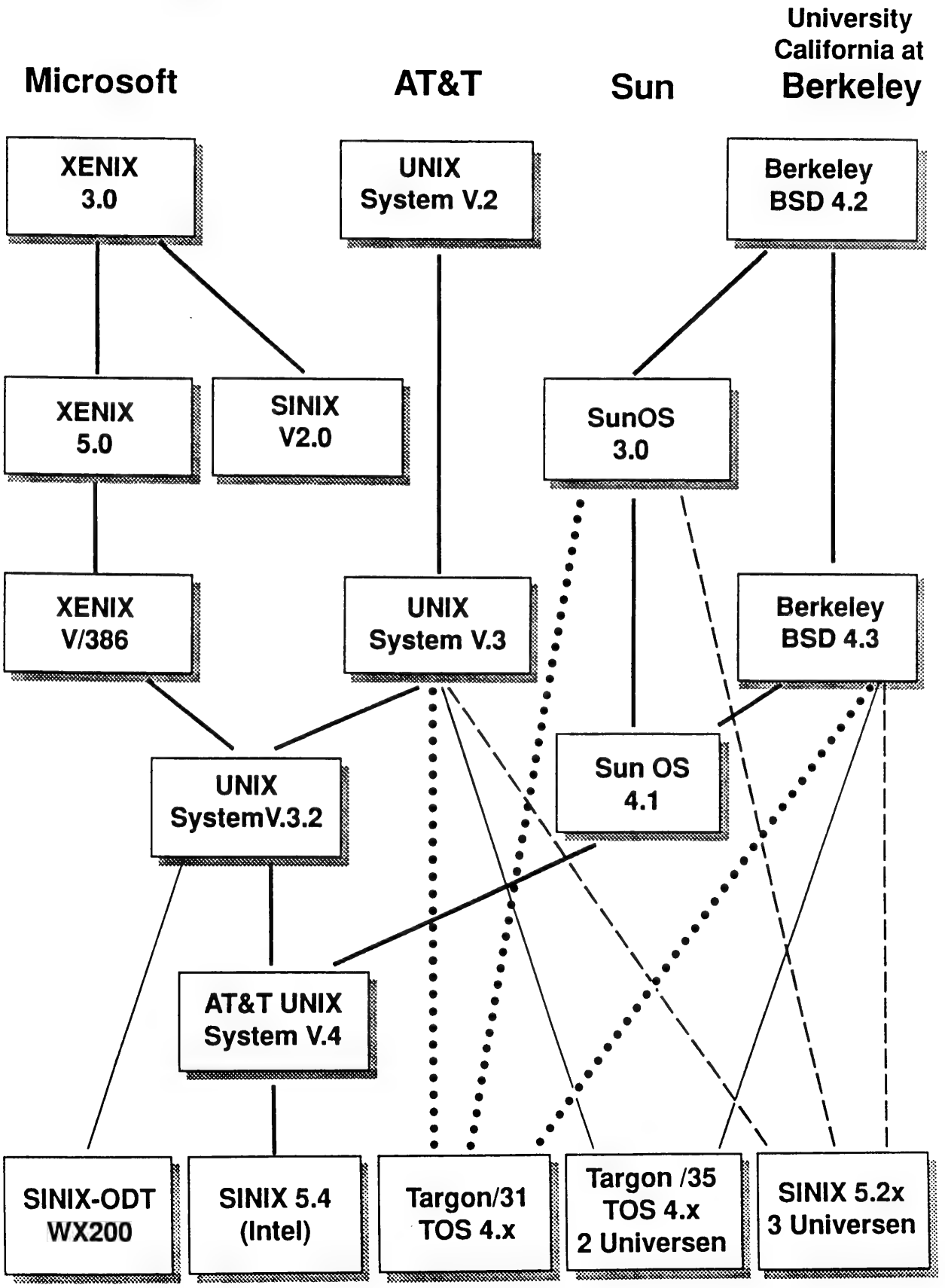
Prozessoren:	INTEL 80486, Taktfrequenz 25 MHz, 2 bis 7 Prozessoren, 4 Mbit-Speicherchips (max. 128 MB Speicher) 128 KB Second Level Cache je Prozessor
Platten:	interne SCSI-Platten am speziellen synchronen SCSI-Bus (4 MB/s) kommerzielle Leistung: max. 84 MIPS (12 MIPS pro Prozessor)

Bus-Systeme:

- Systembus: 52 MB/s
- synchrone SCSI-Kanäle für interne Festplatten: jeweils 4 MB/s
  (2 oder 4 Kanäle im Grundschrank, bis zu 4 für den Erweiterungsschrank)
- asynchroner SCSI-Bus intern: 1,5 MB/s
- bis zu 3 Multibus I für Ein/Ausgabe: jeweils 1,6 MB/s

Achtung: Diese Angaben gelten nur für "echte" MX500-90.
MX500-75, die auf Intel-Prozessor umgerüstet wurden, haben keinen synchronen SCSI-Bus und betreiben ihre Platten weiterhin am Multibus I
(siehe Kap.6 Seite L 6 - 4).

UNIX/SINIX-Stammbaum



X/OPEN-Standards

Die X/OPEN-Standards für UNIX Systeme sind festgelegt im "XOPEN Portability Guide". Es gibt 2 "Portability Guides":

X/OPEN Portability Guide 2: XPG 2 (älterer Standard)

X/OPEN Portability Guide 3: XPG 3 (aktueller Standard)

SINIX V5.4 und SINIX V5.22/V5.23 entsprechen dem aktuellen X/OPEN Portability Guide 3.

Die Kommandos zur Systemverwaltung sind in den X/OPEN-Standards noch nicht festgelegt. Hier unterscheiden sich die SINIX Versionen 5.2x und 5.4 erheblich !

Korn Shell: ksh

Mit SINIX V5.4 wird auch die "ksh" ausgeliefert. Diese Shell ist kompatibel zur normalen "Bourne Shell" ("sh"), bietet aber unter anderem die Möglichkeiten:

- die Kommandozeile mit "vi"-Kommandos zu editieren (wahlweise auch mit "emacs"-Kommandos)

- alte Kommandozeilen erneut zu bearbeiten (history)

Aufruf:

```
ksh -o vi          # Korn Shell mit "vi"-Kommandos
```

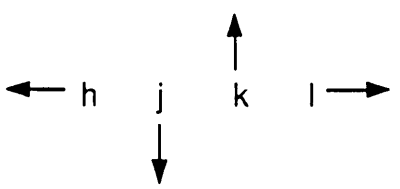
oder

```
ksh                # Korn Shell ohne Editor
set -o vi          # umschalten auf "vi"-Kommandos
```

In Editiermodus umschalten: <ESC>

Editiermodus verlassen: <ENTER> oder <DEL>

Es gelten die original "vi"-Cursor-Tasten:



Im Editiermodus können Sie alle original "vi"-Kommandos verwenden, nicht jedoch die Cursor-Tasten der Tastatur!

Die letzten Kommandozeilen ausgeben: `history`

Die "ksh" numeriert alle Kommandozeilen und speichert sie in der Datei ".sh_history".

Beispiele für history:

25. Kommandozeile editieren: **<ESC>25G**

Kommandozeile editieren,
die mit "prtvtoc" beginnt : **<ESC>/prtvtoc**

Benutzer und Benutzergruppen

Benutzerkennungen:

Der Systemverwalter sollte für jede Person, die am Rechner arbeiten darf, eine eigene Benutzerkennung einrichten.

Prinzipiell könnten mehrere Personen gleichzeitig unter der gleichen Benutzerkennung arbeiten.
Erfahrungsgemäß treten dabei jedoch meist Probleme oder Fehler auf.

⇨ pro Person eine eigene Benutzerkennung !!!

Welche Informationen speichert SINIX für jede Benutzerkennung ?

Benutzergruppen:

Beim Einrichten einer Benutzerkennung müssen Sie angeben, zu welcher Benutzergruppe der neue Benutzer gehören soll.

Bevor Sie eine Benutzerkennung einrichten, sollten Sie daher mindestens eine Benutzergruppe einrichten !!

Die Zugehörigkeit zu einer Benutzergruppe ist wichtig für die Zugriffsrechte auf Dateiverzeichnisse, Dateien und Programme.

Mit den Kommandos **"chmod"** und **"umask"** können Sie dafür sorgen, daß alle Benutzer einer Benutzergruppe auf gemeinsame Dateiverzeichnisse, Dateien und Programme zugreifen können.

Benutzerkennungen zur Systemverwaltung:

root Shellbenutzer mit unbeschränkten Rechten

⇨ ein falsches Kommando kann das gesamte System zerstören

Bestimmte Aufgaben sollte nur "root" erledigen:

- Kontrolle symbolischer Verweise
- System in den "run-level 1" herunterfahren
- Sicherung ganzer Plattenbereiche (Dateisysteme)
- Umkonfigurierung der Festplatte(n)
- Systemuhr stellen

admin Systemverwalter mit menügeführter Benutzeroberfläche (SINIX-Besonderheit):

⇨ Alpha-COLLAGE bzw. Grafik-COLLAGE

Aufgaben (Beispiele):

- Installation von Softwareprodukten
- Benutzergruppen und Benutzerkennungen definieren
- Kennwörter von Benutzern ändern
- Konfiguration von Bildschirmen und Druckern
- zusätzliche Druckergruppen definieren
- Postverteilerlisten verwalten
- LAN- bzw. WAN- Software konfigurieren

sysadm Systemverwalter mit AT&T Menü (englisch)

Was haben diese Benutzer gemeinsam?

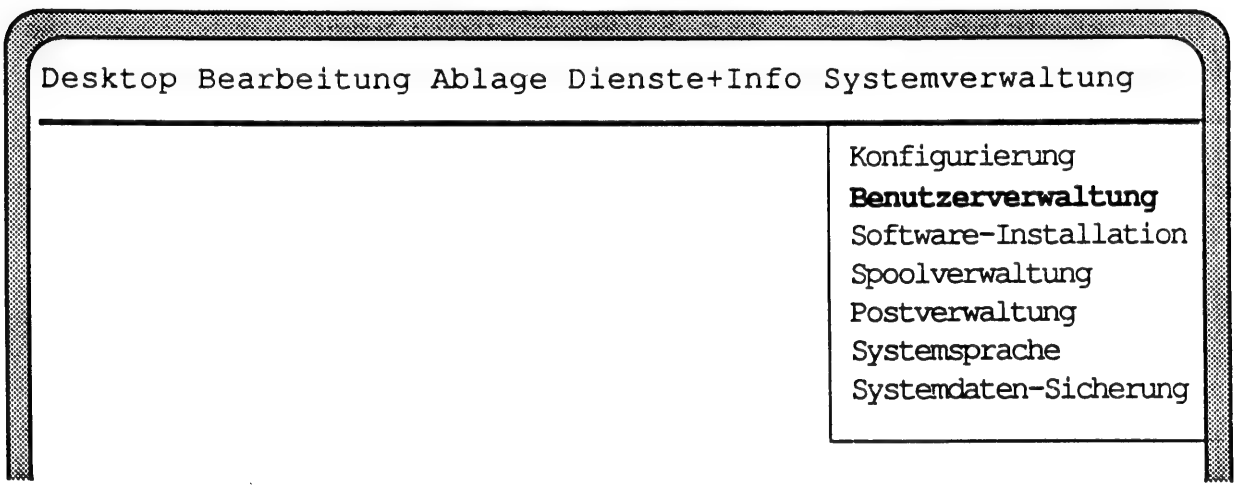
Menügeführte Systemverwaltung

⇨ Benutzerkennung: **admin**

COLLAGE

Drücken Sie die Taste

und verwenden Sie die Cursortasten.
Das Pull-Down-Menü "Systemverwaltung" :



Sie können das Menü nur auswählen, wenn ein Verzeichnisfenster offen ist und "oben" liegt.

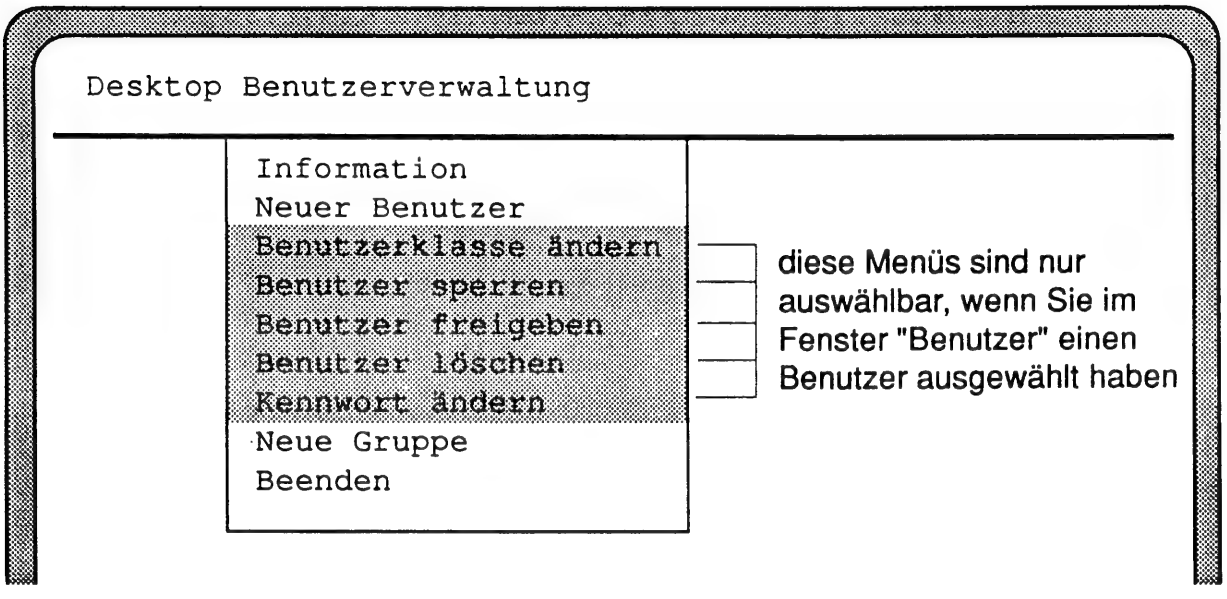
Benutzerverwaltung

Sie können Benutzergruppen und Benutzerkennungen über das "admin" -Bediensystem einrichten, indem Sie aus dem Pull-Down-Menü "Systemverwaltung" auswählen (Cursortasten und):

Benutzerverwaltung

Sie starten damit ein spezielles Programm des Bediensystems. Dieses Programm dürfen Sie nicht von zwei Terminals gleichzeitig verwenden.

Sie erhalten ein Fenster "Benutzer", das eine Liste aller bereits eingetragenen Benutzer enthält. Drücken Sie wieder die Taste MENU . Sie haben eine neue Menüleiste mit folgendem Menü:



Sie können nun den Menüpunkt:

Neue Gruppe

auswählen.

Ist eine Benutzergruppe vorhanden, dann können Sie die Einrichtung eines neuen Benutzers beginnen:

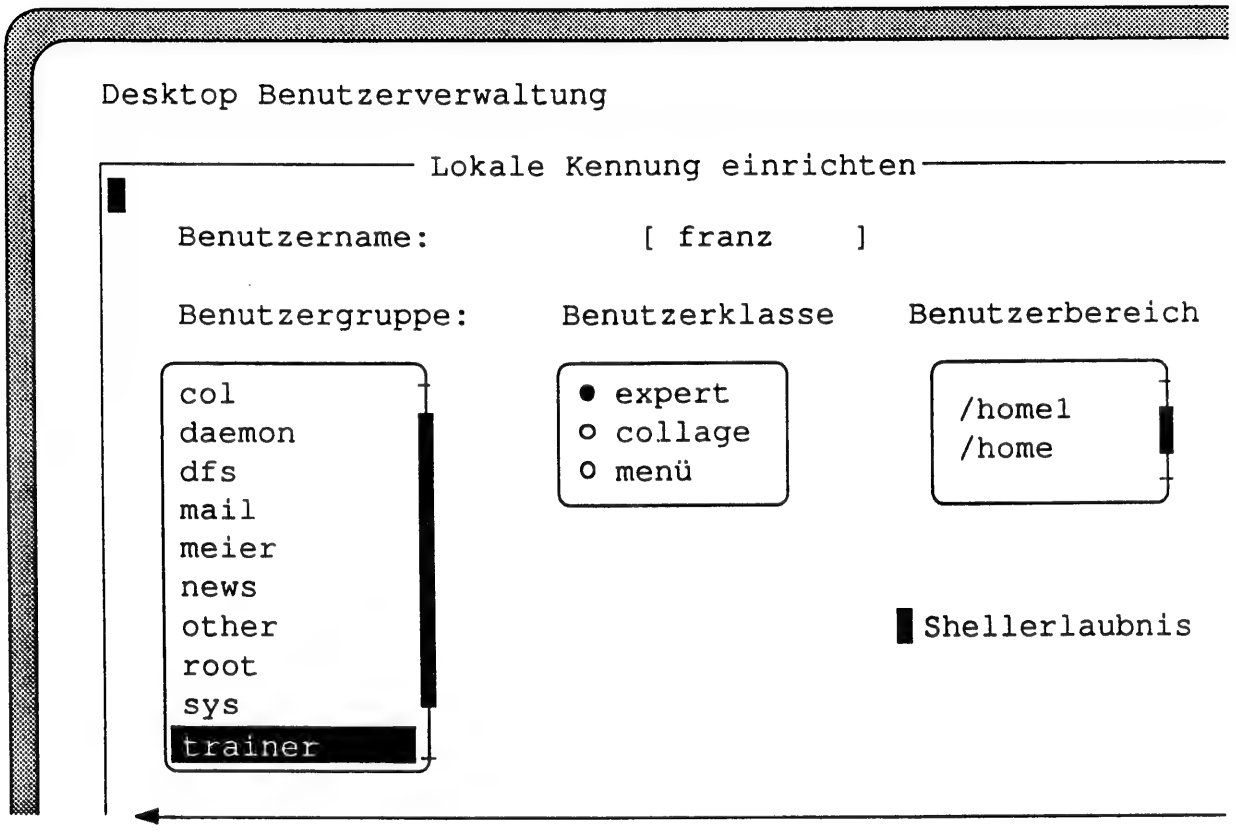
Neuer Benutzer

Sie haben folgende Wahlmöglichkeiten:

- lokale Benutzerkennung (Standardfall)
  ("lokal" = nur an Ihrem Rechner)

- lokale Kennung für globalen Benutzer
  (falls ein Benutzer bereits global im Netz eingerichtet ist)

Neuen Benutzer eintragen



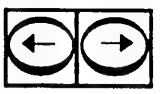
Mit den Tasten oder können Sie sich im Fenster bewegen.

Benutzerklasse:	- expert	Shellbenutzer
	- collage	COLLAGE-/bzw. OCIS-Benutzer
	- menü	altes Menüsystem
Desktops:	- collage	Benutzeroberfläche
	- ocis2	
	...	

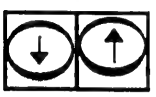
Benutzerbereich (Dateisystem): /home, /home1, ...

Wichtige Tasten von Alpha-COLLAGE

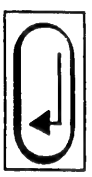
Pull-Down-Menüs herunterklappen



ein anderes Menü auswählen



Auswahl in einem Menü oder in einem Auswahlfenster



ein Objekt auswählen oder Auswahl an COLLAGE senden

zur nächsten Objektgruppe / Auswahl gehen
(wichtige Tasten für Alpa-COLLAGE)

die aktuelle Aktion abbrechen

ausgefülltes Fenster abschicken

Betriebsart

Betriebsart wechseln:
Alpha-COLLAGE besitzt 2 Betriebsarten
- Betriebsart "Anwendungen" (Standard)
- Betriebsart "System" (für Sonderfälle)

Sie erkennen die aktuelle Betriebsart an der Form des Cursors:

▮ Cursor für Betriebsart "Anwendungen"
In dieser Betriebsart werden fast alle Tasteneingaben an die aktive Anwendung weitergeleitet.

\+ Cursor für Betriebsart "System"
Der Cursor wird anstelle eines Mauszeigers verwendet.
Alle Funktions- und Sondertasten werden von Alpha-COLLAGE abgefangen

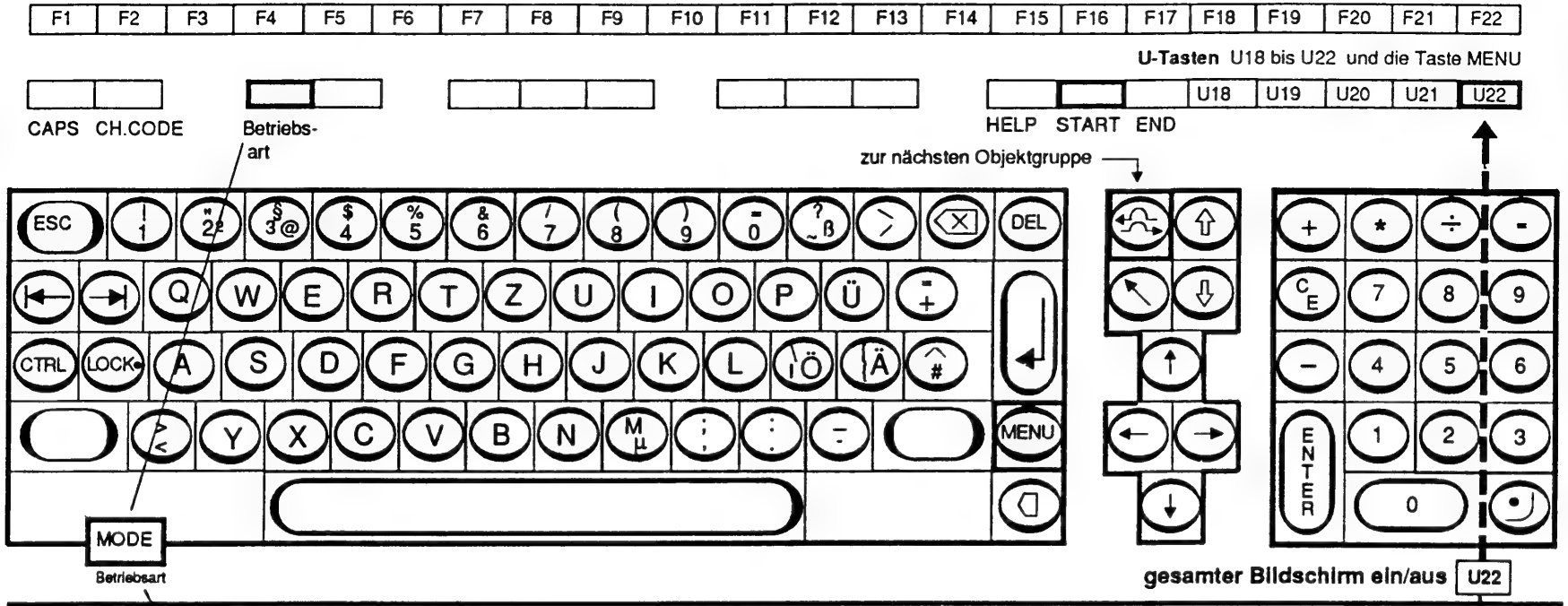
Wichtige Sondertasten:

gesamter Bildschirm ein/aus

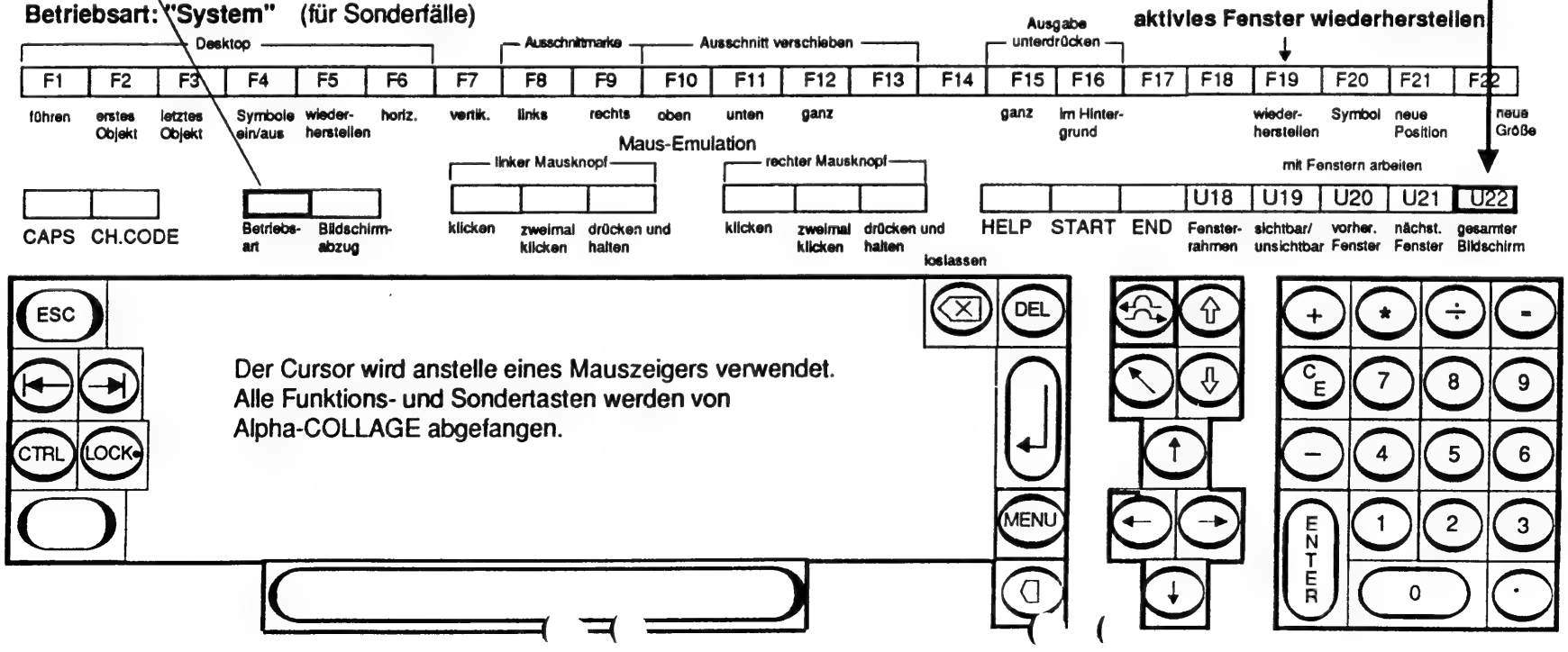
aktives Fenster wiederherstellen

Betriebsart: "Anwendungen" (Standard)

In dieser Betriebsart werden fast alle Tasteneingaben an die aktive Anwendung weitergeleitet.



Betriebsart: "System" (für Sonderfälle)



Der Cursor wird anstelle eines Mauszeigers verwendet.
Alle Funktions- und Sondertasten werden von Alpha-COLLAGE abgefangen.

/etc/passwd, /etc/shadow

/etc/passwd: (lesbar für alle)

```
root:x:0:1:0000-Admin(0000):/:
daemon:x:1:1:0000-Admin(0000):/:
bin:x:2:2:0000-Admin(0000):/usr/bin:
            ...
sync:x:67:1:0000-Admin(0000):/:/usr/bin/sync
tele:x:90:1:0000-teleservice(0000):/home/tele:
install:x:101:1:Initial Login:/home/install:
nobody:x:100:101:unprivileged user:/nonexistent:/noshell
admin:x:0:2:Administrator mit COLLAGE:/usr/admin:/usr/bin/collage.pw
sysadm:x:0:0:AT&T administration:/usr/admin:/usr/sbin/sysadm
            ...
franz:x:104:103::/home/franz:/sbin/sh
meier:x:105:103:Rudi Meier:/home2/meier:/sbin/sh
```

Verweise auf /etc/shadow
Gruppennummer
Benutzer-nummer
wahlweise der vollständige Name
Login-Dateiverzeichnis
Startprogramm

/etc/shadow: (Verwaltung der Kennwörter)

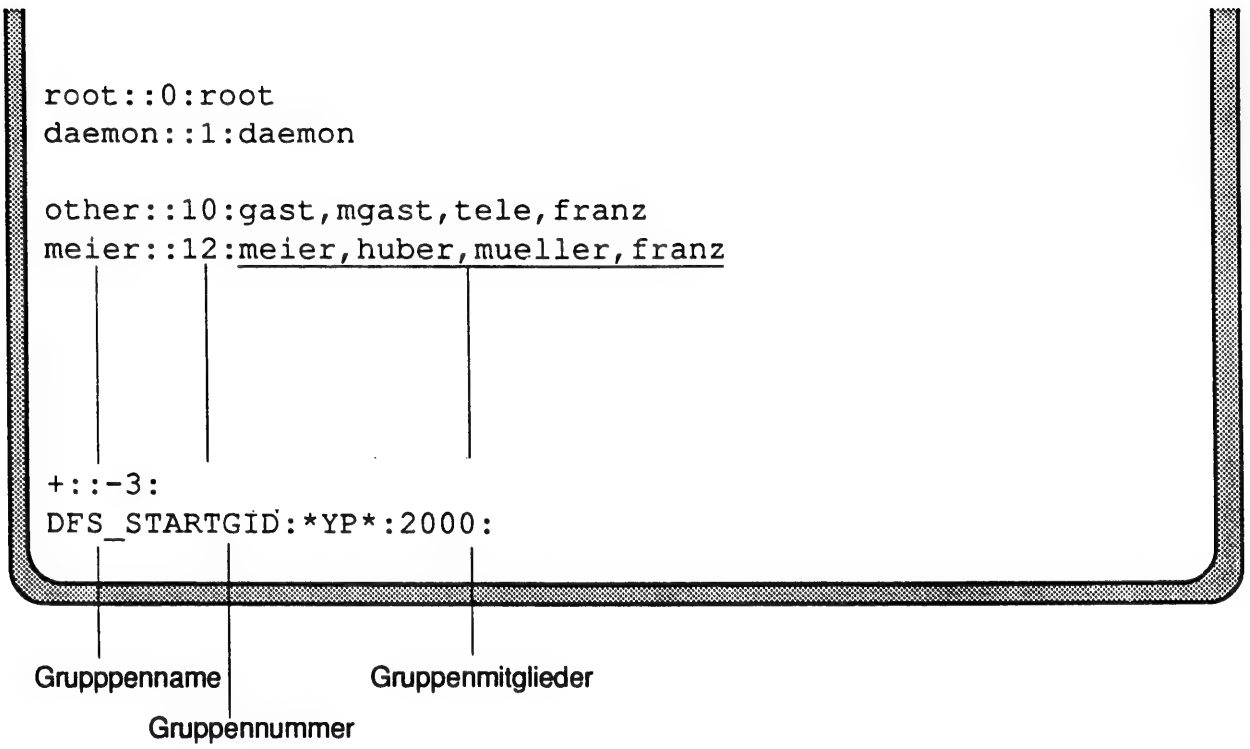
```
root:nn61ZTDxP0omo:7601:0:168:7:::
daemon:NONE:7601::::::
bin:NONE:7601::::::
            ...
sync::7601::::::
tele:NoCpAmPZiKyg2:7601:0:168:7:::
install:4o0Dfy3aYkECc:7601:0:168:7:::
nobody:*LK*:::::::
admin:2xno2kDQfcxkI:7602:0:168:7:::
sysadm:dtpaAwYdCc/sQ:7602:0:168:7:::
            ...
franz:NPkbOD/p/Xg9.:7624:0:168:7:::
meier:muNrPZEQx4xYs:7655:0:168:7:::
```

- verschlüsseltes Kennwort
- gesperrt: NONE, *LK*
- kein Kennwort (leeres Feld)

Tag der letzten Kennwort-Änderung (seit 01.01.1970)
Minimumzeit (in Tagen)
Maximumzeit
Warnzeit

/etc/group

/etc/group: Gruppenname <—> Gruppennummer (GID)



Sie können Benutzer zusätzlich als Mitglied in mehrere Benutzergruppen eintragen. Der Benutzer kann dann seine Gruppe wechseln mit dem Kommando:

newgrp Gruppe

Kommandos zur Benutzerverwaltung

```
                        # neue Benutzergruppe einrichten
groupadd GRUPPE

                        # Benutzergruppe löschen
groupdel GRUPPE
```

```
                        # neuen Benutzer einrichten
useradd  -g GRUPPE  -d /home/meier  -m  meier
```

Gruppe — HOME-Verzeichnis — HOME-Verzeichnis anlegen — Benutzer-name

Im Anschluß an "useradd" ist der Benutzer gesperrt. Sie müssen als nächstes ein Kennwort für den neuen Benutzer vergeben:

```
passwd meier
```

—> Kennwort eingeben

Weitere Schalter von "useradd":

`-s /usr/bin/ksh`	Startprogramm
`-c "Dr. Hans Meier"`	Kommentar

```
                        # Benutzer mit Home-DVZ löschen
userdel -r meier
```

Die Kommandos sind nicht verwendbar für Menü/(MES)-Benutzer !

Das "passwd"-Kommando

Mit dem "**passwd**"-Kommando können Sie Ihr eigenes Kennwort ändern.
Der Systemverwalter kann mit dem "**passwd**"- Kommando:

- Kennwörter und Verwaltungsinformationen beliebiger Benutzer ändern
- Verwaltungsinfomationen aller Benutzer ansehen

passwd -s benutzer

zeigt den Status des Kennworts und die Zeit-Angaben aus der "**/etc/shadow**"

Status: PS — Kennwort ist vorhanden
LK — Kennung gesperrt oder DFS Kennung
NP — kein Kennwort vorhanden

passwd -as # nur Systemverwalter

zeigt die obigen Angaben für alle Benutzer

passwd -l benutzer

sperrt die Benutzerkennung

passwd -x max_tage -n min_tage benutzer

legt die maximale und die minimale Gültigkeitsdauer in Tagen fest

opasswd und oshadow

Bei jeder Änderung in der "**passwd**" oder der "**shadow**"- Datei sichert das "**passwd**"-Kommando den alten Stand in der "**/etc/opasswd**" beziehungsweise "**/etc/oshadow**".

/etc/default/login, /etc/default/su

In der Datei "**/etc/default/login**" können Sie Parameter für jedes Login setzen:

Beispiel:
```
HZ=100
ULIMIT=32000
CONSOLE=/dev/term/tty000
PASSREQ=YES
ALTSHELL=YES
PATH=/usr/bin:/usr/sbin:/opt/bin:.
SUPATH=/usr/bin:/usr/sbin:/opt/bin:/sbin:/opt/etc:/usr/ucb
```

Bedeutung der Variablen:

CONSOLE Falls dieser Parameter gesetzt ist, kann sich der "**super-user**" nur über das angegebene Terminal anmelden. Ist der Parameter nicht gesetzt, kann sich der "**super-user**" an jedem Terminal anmelden.

PASSREQ=YES alle Benutzer müssen ein Passwort haben (bis auf "**root**")

ALTSHELL=YES die Variable SHELL darf einen anderen Wert annehmen als "**/bin/sh**"

SUPATH PATH für Logins mit der Benutzernummer 0
Dieser PATH gilt nicht für das "**su**"-Kommando.

In der Datei "**/etc/default/su**" können Sie Parameter für das "su"-Kommando setzen:

Beispiel:
```
SULOG=/var/adm/sulog
CONSOLE=/dev/term/tty000
SUPATH=/usr/bin:/usr/sbin:/opt/bin:/sbin:/opt/etc:/usr/ucb
```

CONSOLE Falls dieser Parameter gesetzt ist, gibt das "**su**"-Kommando bei jedem "**root**"-su eine Meldung am angegebenen Terminal aus.

SUPATH PATH nach "**root**"-su

Weitere Kommandos zur Systemverwaltung:

chmod Zugriffsrechte verändern

```
chmod ug+rw ....
chmod ug+x ....
```

chown Eigentümer verändern (nur root !)

```
chown meier ....
```

In SINIX VS.5 besitzen die Kommandos "chmod", "chown" und "chgrp" einen neuen Schalter "-R" (rekursiv). Dieser Schalter bewirkt, daß die Kommandos auch alle Unterverzeichnisse bearbeiten:
`chown -R meier /home/meier`

chgrp Gruppe verändern (nur root !)

```
chgrp gruppe4 ....
```

umask Zugriffsrechte für neue Dateien verbieten (vgl. /etc/profile)

```
umask 027          # Gruppenmitglieder dürfen nicht schreiben,
                   # andere dürfen weder lesen noch schreiben
```

Eigentümer — Gruppe — "andere"

1 == ausführen
2 == schreiben
4 == lesen
verbieten

#Jedes Bit, das Sie bei "umask" setzen,
ist bis zum Ende der aktuellen Shell für
#neuangelegte Dateien verboten.

du -s belegten Plattenplatz feststellen (in 512 Byte-Blöcken)

```
du -s /home/mueller
```

df -k Plattenbelegung des Gesamtsystems (in KB)

whodo, **finger** Informationen über Aktivitäten der angemeldeten Benutzer

Kommandos, Tagesmeldung, Eröffungsbildschirm

wall Meldung an alle Terminals senden (Kommando für Systemverwalter)

"ps" -Kommando:

Liste aller Prozesse:	`ps -ef | pg`
alle eigenen Prozesse auflisten :	`ps -fu $USER`
alle Prozesse von Terminal tty005:	`ps -ft tty005`
Prozeßgrößen kontrollieren: (Spalte SZ in 4KB-Seiten)	`ps -el | pg`

kill PID Prozeß "sanft" beenden (Signal 15)

kill 0 alle Hintergrundprozesse des eigenen Terminals sanft beenden (funktioniert nicht bei COLLAGE)

kill -9 PID Prozeß "brutal" beenden (Signal 9)

kill -9 0 alle Prozesse des eigenen Terminals brutal beenden (nicht bei COLLAGE)

Tagesmeldung, Eröffnungsbildschirm:

/etc/motd Datei mit Tagesmeldung (muß für alle lesbar sein und wird bei jedem Anmelden ausgegeben)

/etc/issue Eröffnungsbildschirm für alle Terminals (muß für alle lesbar sein)

t-Bit bei "/tmp" und "/var/tmp"

- Unter welchen Voraussetzungen kann ein normaler Benutzer fremde Dateien löschen ?

- Was bedeutet dies normalerweise für Dateien unter /tmp, /usr/tmp und anderen allgemeinen Verzeichnissen ?

- Prüfen Sie die Zugriffsrechte von /tmp in SINIX V5.4.
  Versuchen Sie, fremde Dateien zu löschen!

- Probieren Sie aus:

```
cd $HOME
mkdir ALLE
chmod 777 ALLE
chmod +t ALLE
echo hallo > ALLE/datei
```

Bitten Sie nun Ihre Nachbargruppe, eine Datei "datei2" in Ihrem DVZ anzulegen und Ihre Datei zu löschen.

Aufgabe 1:

a) Sie wollen als Systemverwalter das Kennwort eines Benutzers ändern. Welche Möglichkeiten haben Sie ?

passwd benutzer

b) Sie übernehmen die Systemverwaltung eines vom Werk installierten SINIX-Rechners. Bei welchen Benutzerkennungen sollten Sie sofort die Kennwörter ändern ?

Root, Admin, SU

c) Ihr Kursleiter teilt Ihnen mit, wie Sie das COLLAGE-Bediensystem starten können:

```
acollage
```

Machen Sie sich mit der Bedienung dieser Benutzeroberfläche vertraut. Sie sollten jedoch nur Aktionen starten, die für alle Benutzer erlaubt sind.

Verwenden Sie für die folgenden Aufgaben die "ksh" und nutzen Sie die Editiermöglichkeiten.

d) Stellen Sie fest:

- wieviel Plattenplatz die Daten in Ihrem HOME-Verzeichnis belegen ("du"-Kommando) 142 Block zu 512 k
- wieviel Platz auf der Platte noch frei ist 409 MB
- mit welchen Kommandos/Programmen die anderen Kursteilnehmer im Moment arbeiten

find, ksh, collage

Fortsetzung von Aufgabe 1:

e) Ändern Sie die Zugriffsrechte aller Dateien und Datei-Verzeichnisse unterhalb Ihres Login-Verzeichnisses:

Erweitern Sie zunächst die Zugriffsrechte für Ihre Gruppe und zwar:

rw—	für Dateien
rwx—	für Dateiverzeichnisse

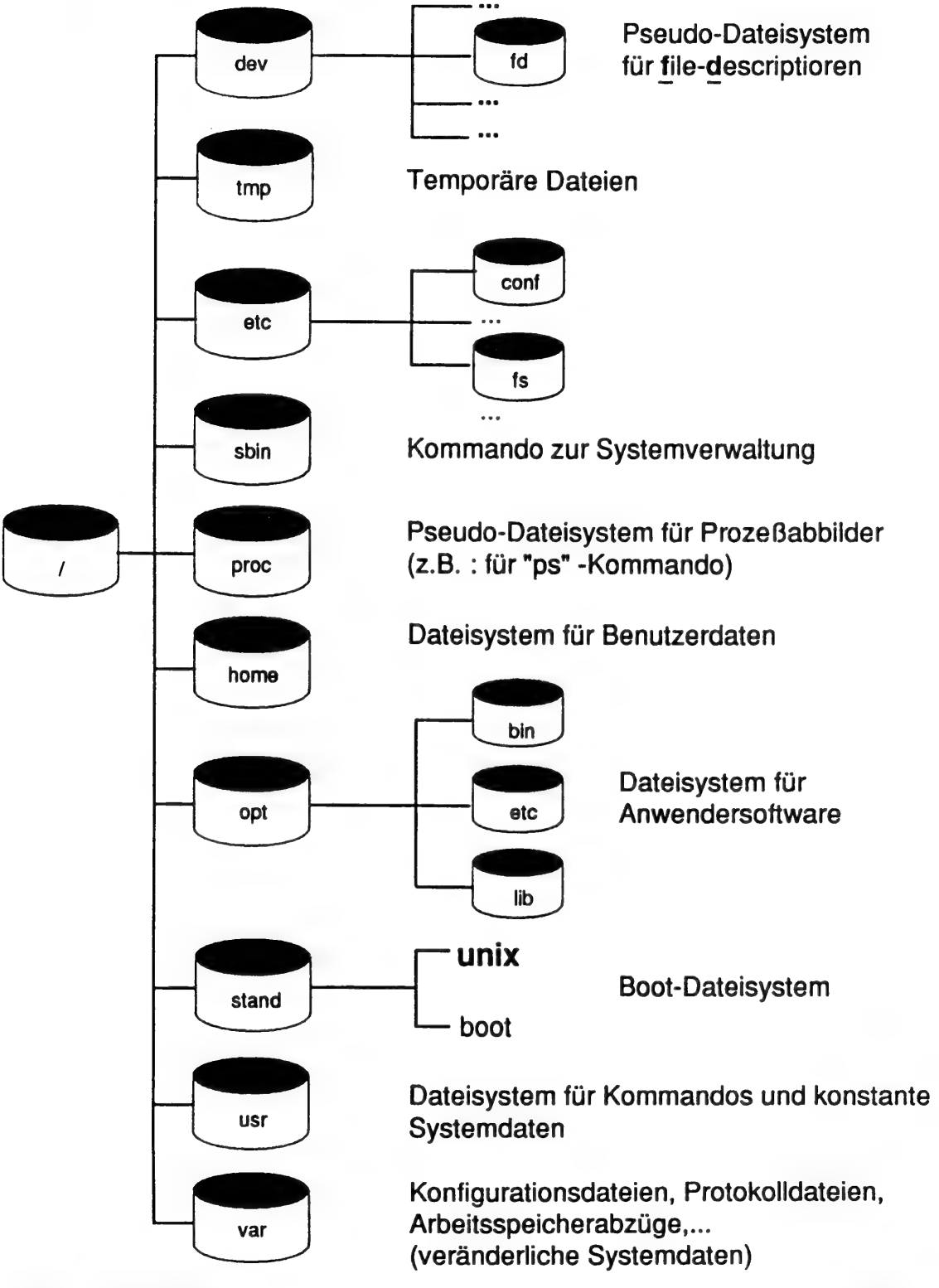
Überprüfen Sie das Ergebnis mit "ls -lR".

f) Sorgen Sie dafür, daß Ihre Gruppenmitglieder alle von Ihnen in Zukunft neu angelegten Dateien lesen können!

e) find . -type f -print | xargs chmod g+rw
find . -type d -print | xargs chmod g+rwx

f) umask 037

Dateibaum in SINIX V5.4/UNIX V.4



Aus Kompatibilitätsgründen wurden die alten Dateiverzeichnisse "**/bin**", "**/lib**", ... durch einfache symbolische Verweise auf die neuen Dateiverzeichnisse ersetzt.

Dateibaum in SINIX V5.4

/dev	Gerätedateien
/dev/fd	Pseudo-Dateisystem für file-descriptoren
/export	Montierpunkt für Dateisysteme, die von anderen Rechnern über ein Netz erreichbar sein sollen
/etc	Systemdateien, **keine** Kommandos
/etc/fs/...	Kommandos für verschiedene Dateisystemtypen
/etc/conf/..	Dateien, um einen Systemkern zu binden
/sbin	Kommandos und Programme zur Systemverwaltung
/home **/home1** **...**	eigene Dateisysteme für Benutzerverzeichnisse z.B: /home/meier, /home1/franz
/install	für Installation von Softwarepaketen
/opt	Dateisystem für Anwendersoftware
/proc	Pseudo-Dateisystem für Prozeßabbilder (kein Platzbedarf)
/shlib	aus Kompatibilität zu UNIX System V.3
/stand	Boot-Dateisystem mit Systemkern (ca. 5 MB)
/tmp	Dateiverzeichnis/Dateisystem für temporäre Dateien
/usr	Kommandos und konstante Systemdaten (/usr/bin, /usr/sbin, /usr/ucb, /usr/ccs, ...)
/var	Dateiverzeichnis/Dateisystem für Dateien, die laufend geändert werden (Konfigurations-,Protokolldateien, Arbeitsspeicherabzüge, ...)

Das "/var"-Dateisystem

/var/adm/	Accounting, Protokolldateien astlog, sulog, utmp*, wtmp*
/var/adm/klog.msg	Meldungen des Systemkerns
/var/crash/	**Arbeitsspeicherabzüge des Systems**
/var/sadm/	Installationsprotokolle, eventuell Backup-Protokolle
/var/ced/	Hilfsdateien für den ced-Editor
/var/mail/	Verzeichnis für Postkörbe
/var/cron/log	Protokolldatei des "cron"
/var/spool/cron/	Dateien und Dateiverzeichnisse für "crontab" und für "at"-Kommando
/var/spool/spooler/	für SINIX-Spoolsystem
/var/tmp/	temporäre Dateien (früher: "/usr/tmp")
/var/saf/	Protokolldateien für Netzsoftware
/var/yp/	für "yellow pages" (Netzsoftware)

Das "/opt"-Dateisystem (40 - 100 MB)

Hier soll alle Anwendersoftware (optionale Software) installiert werden. Bei SINIX V5.4 wird das SINIX-Spoolsystem ("lpr") unterhalb von "/opt" installiert.

/opt/bin/	Programme, Kommandos z.B.: lpr, maxed
/opt/etc/	z.B.: interne Programme des SINIX-Spools
/opt/lib/ /opt/include/	Hilfsdateien/Dateiverzeichnisse
/opt/emds/ /opt/ft/	produktspezifische Verzeichnisse

Das "/usr"-Dateisystem (60 - 70 MB)

Hier sollten alle Systemprogramme und Dateien stehen, die sich nicht ändern.
Leider wird dieses Konzept noch nicht konsequent verfolgt.

/usr/admin/

/usr/bin/	**fast alle Kommandos**
/usr/sbin/	**zusätzliche Kommandos zur Systemverwaltung**
/usr/ccs/	original C-Entwicklungssystem von AT&T: normale Bibliotheken: lib*.a "shared libraries": lib*.so
/usr/ces/	SINIX C-Compiler (verwendet Bibliotheken aus "/usr/ccs/lib")
/usr/include/	
/usr/lib/	Hilfsdateien, normale Bibliotheken: lib*.a "shared libraries": lib*.so und : libc.so.1
/usr/share/	Dateien, auf die mehrere Rechner im Netz zugreifen dürfen (z.B. ONLINE-Manuale)
/usr/share/lib/terminfo	Terminfo-Verzeichnisse und Dateien
/usr/sadm/	für "sysadm", **nicht verwechseln mit "/var/sadm"**

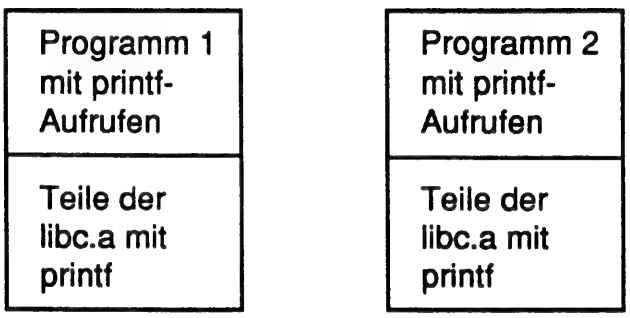
Falls Sie bei der Installation des Betriebssystems das Paket "**compat**" installiert haben, existieren die folgenden "Berkeley" Verzeichnisse:

/usr/ucb/	eine Auswahl von Berkeley-Kommandos z.B.: "mt", "ucb-ps", ...
/usr/ucbinclude/ /usr/ucblib/	C-Entwicklungsumgebung von Berkeley

Shared Libraries in V5.4

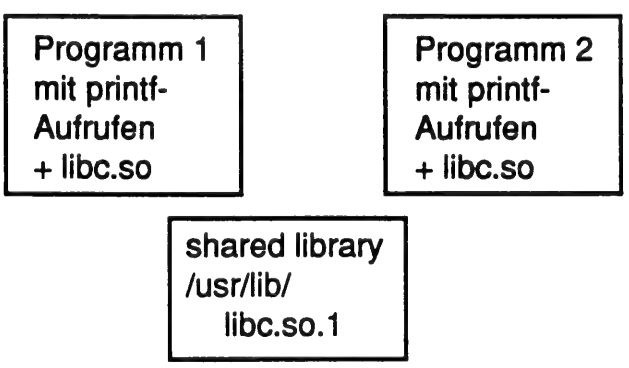
Statisch übersetzte Programme:

```
cc -o programm1 -dn programm1.c
cc -o programm2 -dn programm2.c
```



Programme, die dynamische "shared libraries" nutzen:

```
cc -o programm1 programm1.c
cc -o programm2 programm2.c
```



Die Programme werden gebunden mit "/usr/ccs/lib/libc.so" und sind nur lauffähig, wenn Sie zur Laufzeit auf "/usr/lib/libc.so.1" zugreifen können (Zugriff: r-x l).

Mit dem Kommando **ldd** können Sie prüfen, welche shared libraries zur Laufzeit von einem Programm benötigt werden:

```
ldd /usr/bin/ced

ldd /sbin/fsck

ldd /usr/bin/X11/mwm          # falls Motif installiert ist
```

Einfache symbolische Verweise:

Ein einfacher symbolischer Verweis ("symbolic link") ist eine spezielle Datei.
Sie enthält den Pfadnamen einer anderen Datei oder eines Dateiverzeichnisses.

Beispiele:

/bin ———> /usr/bin
/lib ———> /usr/lib

Aufgabe 2:

In SINIX V5.4 sind die meisten Kommandos unter anderen Pfadnamen erreichbar als in früheren SINIX / UNIX Versionen.

Aus Kompatibilitätsgründen wurden symbolische Links eingerichtet, so daß die Kommandos scheinbar unter den früheren Pfadnamen ansprechbar sind.

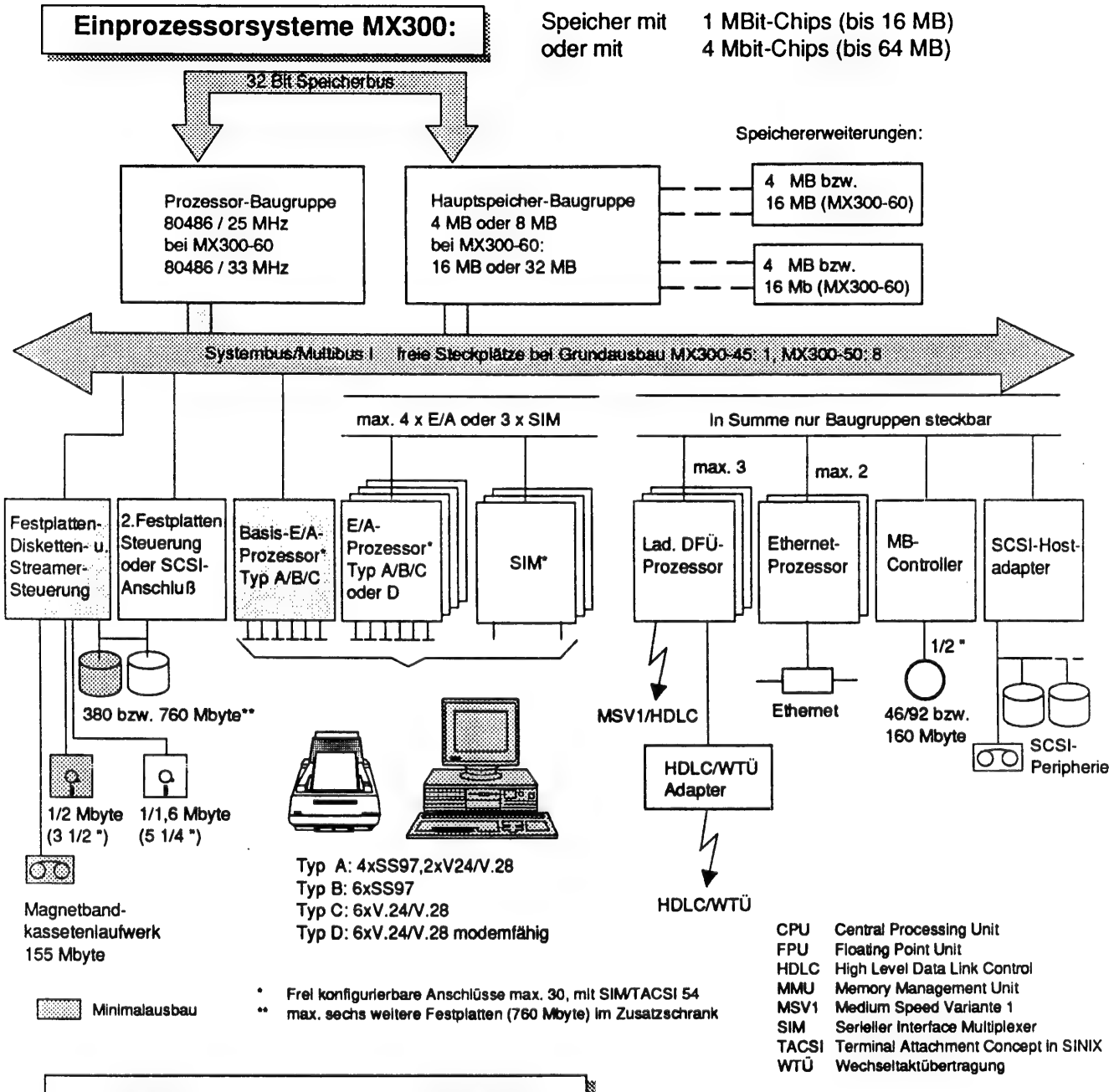
Prüfen Sie die Pfadnamen folgender Kommandos:

- wo stehen die Kommandos in SINIX V5.4 ?
- sind die Kommandos über symbolische Links auch unter anderen Pfadnamen erreichbar ?

Kommando/ Programm	tatsächlicher Pfadname:	symbolische Verweise
sh	/sbin/sh	/bin/sh
ls	/bin/ls	
tar	/usr/sbin/tar	/bin/tar
ps	/sbin/ps	/bin/ps
passwd	/bin	

Hardware, Plattenbereiche und Dateisysteme

Interner Aufbau der SINIX-Rechner



Festplattencontroller und Festplatten:

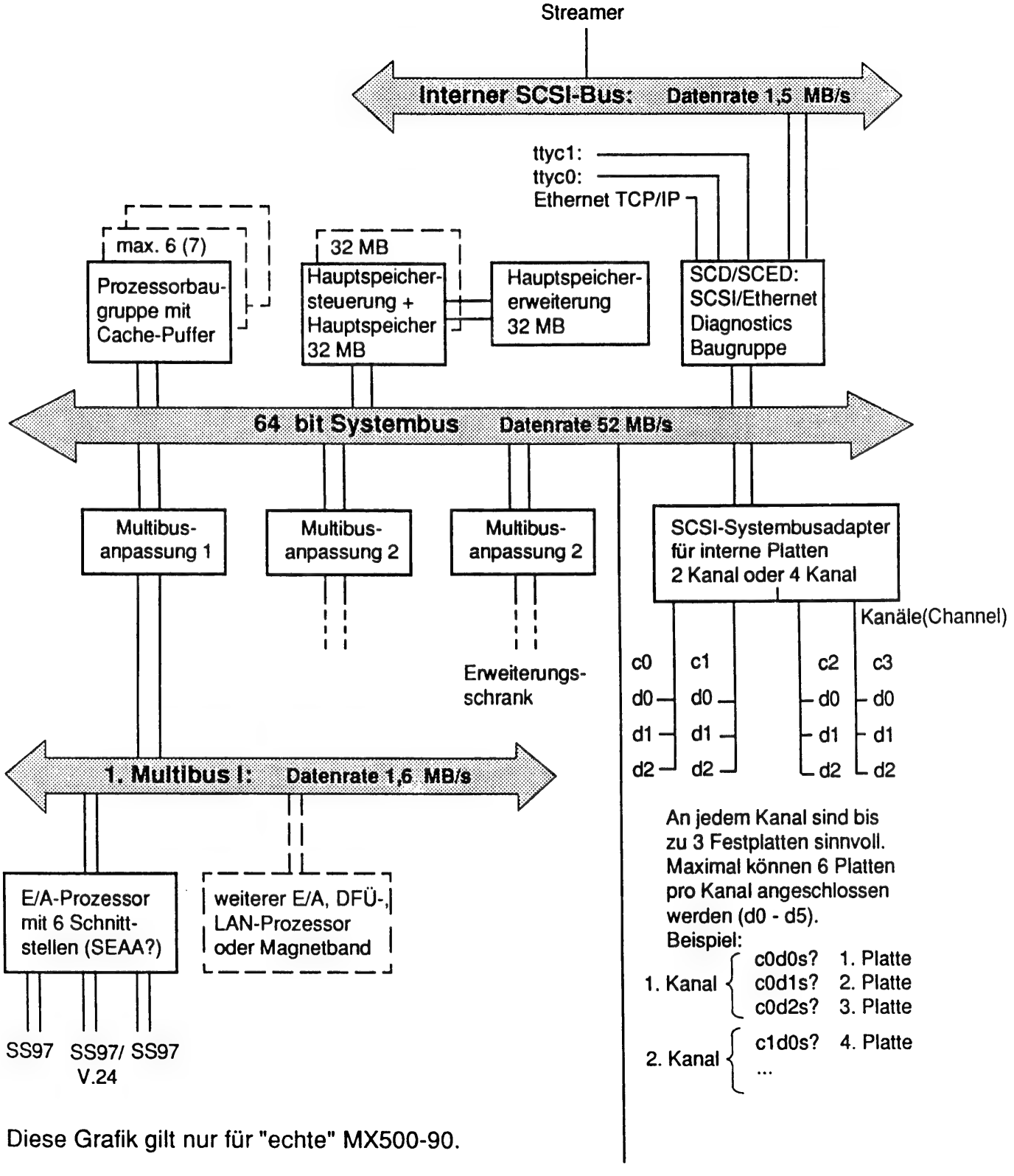
("Controller" == Plattensteuerung)

Bis zu 2 interne Platten mit ESDI-Steuerung:

- entweder beide Platten an der 1. Plattensteuerung
- oder 2. Platte an einer eigenen Plattensteuerung

Externe SCSI-Platten sind über eine spezielle SCSI-Steuerung anschließbar.

Parallelprozessorsystem MX500-90



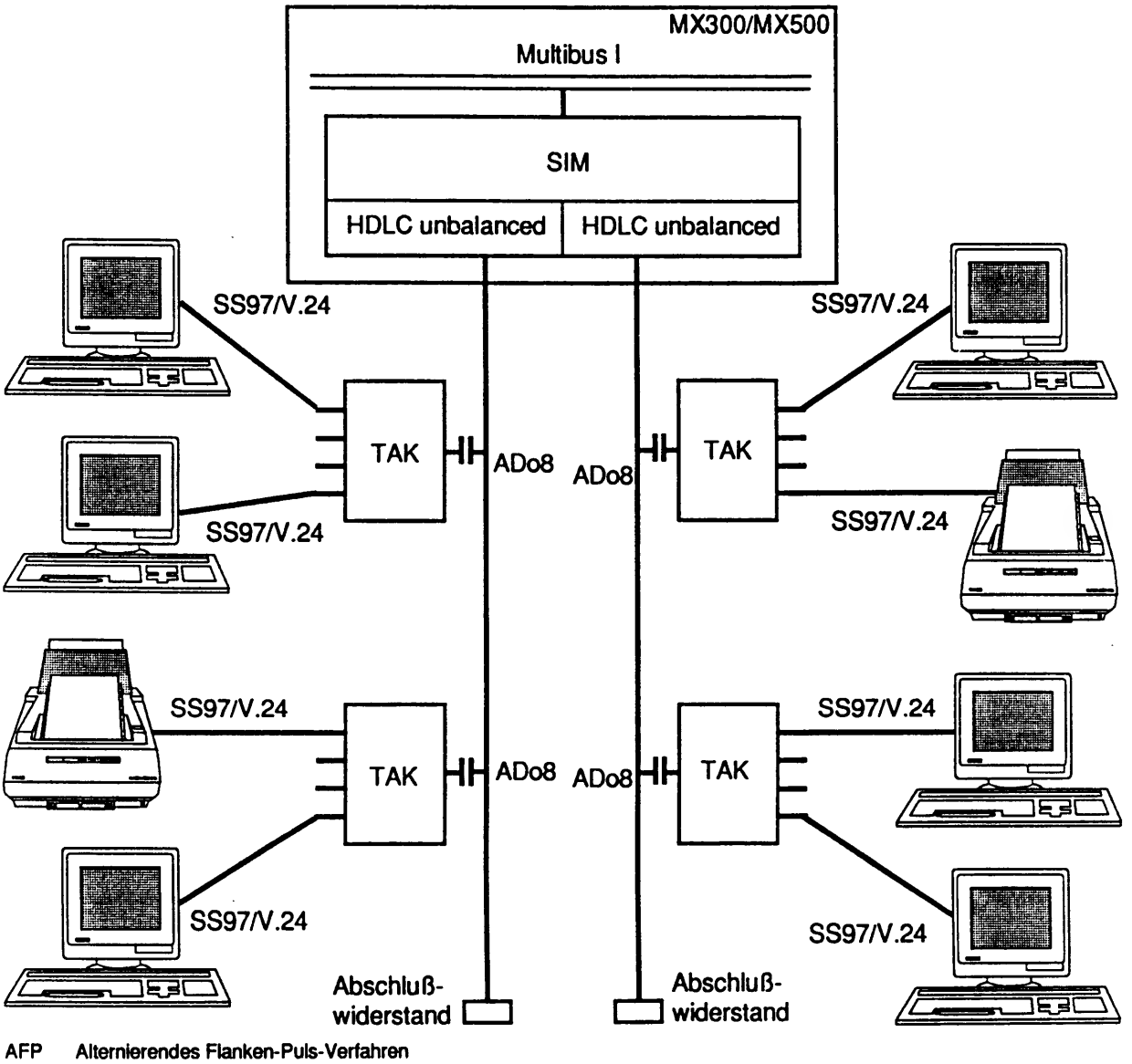
Diese Grafik gilt nur für "echte" MX500-90.

TACSI

TACSI (Terminal Anschluß Konzept in SINIX) verwendet anstelle der E/A-Prozessoren spezielle "SIM"-Multiplexer-Baugruppen (SIM == Serial Interface Multiplexer).

Jede SIM-Baugruppe unterstützt bis zu 16 Schnittstellen. Bis zu 8 Schnittstellen werden vom Multiplexer auf eine AFP-2-Draht-Leitung mit HDLC-Prozedur (187,5 Kbit/s) geleitet.

An jede AFP-2-Draht-Leitung können Sie 2 TAK-Geräte (TAK == Terminal Anschluß Konzentrator) mit maximal 4 Schnittstellen anschließen:

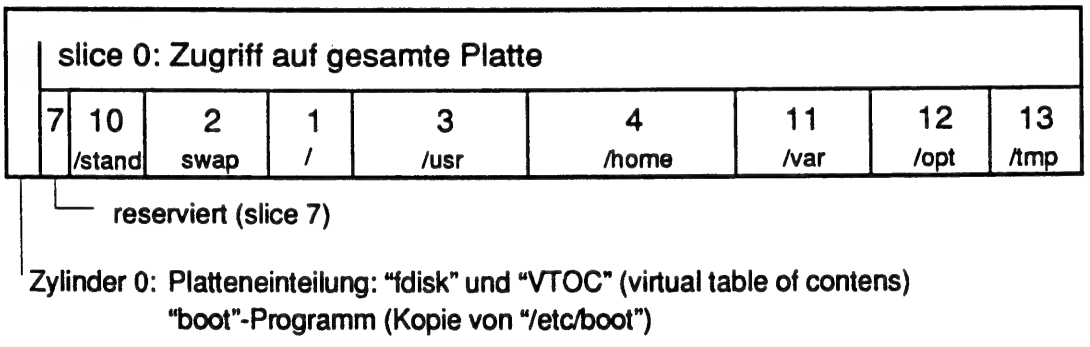


AFP Alternierendes Flanken-Puls-Verfahren
ADo8 Anschlußdose ADo8
TAK Terminal-Anschluß-Konzentrator
SIM Serial Interface Multiplexer für Multinus I
HDLC High-Level Data Link Control

Logische Platteneinteilung

Eine Festplatte ist unterteilt in bis zu 15 Bereiche, die SINIX als getrennte "logische Plattenlaufwerke" behandelt.
Bezeichnung: Plattenbereich oder "slice" 0 bis 15

Beispiel:



Die Größe der einzelnen Bereiche oder "slices" können Sie bei einer Neu-Installation des Betriebssystems festlegen.

Aktuelle Platteneinteilung lesen:

Die Platteneinteilung jeder Platte steht im "VTOC".
Der Systemverwalter kann diese Tabelle über die Gerätedatei für "slice 0" lesen:

```
prtvtoc -f vtoc1  /dev/rdsk/c0d0s0
cat vtoc1     # 1.Platte

prtvtoc -f vtoc2  /dev/rdsk/c0d1s0 ||
                  prtvtoc -f vtoc2/dev/rdsk/c1d0s0
                                  oder
cat vtoc2 # 2.Platte
```

Die Datei /etc/partitions

Die Platteneinteilung aller Platten sollte zusätzlich in der Datei "/etc/partitions" hinterlegt sein.

Hardware, Plattenbereiche und Dateisysteme

Sie können die Bereiche der Festplatte(n) (die "logischen Plattenlaufwerke") für verschiedene Zwecke nutzen:

- **"slice"** 2 der 1.Platte ist automatisch **"swap"**-Bereich
- In anderen Bereichen erzeugen Sie normalerweise **Dateisysteme,** die Sie in Ihren Dateibaum einhängen:

Dateisysteme in SINIX V5.2x:		SINIX V5.4:
"/"	"root"-Bereich: Systemprogramme und Systemdaten	"/" "/stand" mit unix Kern
"/usr"	System- und Anwendersoftware "/opt"	"/usr" "/var" "/tmp"
"/usr1" "/usr2" ...	Benutzerdaten	"/home" "/home1" ...

- Falls Sie Turbo-Informix bzw. Informix-Online verwenden, reservieren Sie weitere Plattenbereiche exklusiv für Ihre Datenbanken und konfigurieren dort kein Dateisystem.

Die swap-Bereiche:

- Auslagerungsbereiche (für **"paging"** und **"swapping"**), die der Betriebssystemkern unbedingt benötigt
- Sie können mehrere **"swap"**-Bereiche konfigurieren
- **ist die Summe der "swap"-Bereiche zu klein, bleibt der Rechner bei Belastung "stehen"**

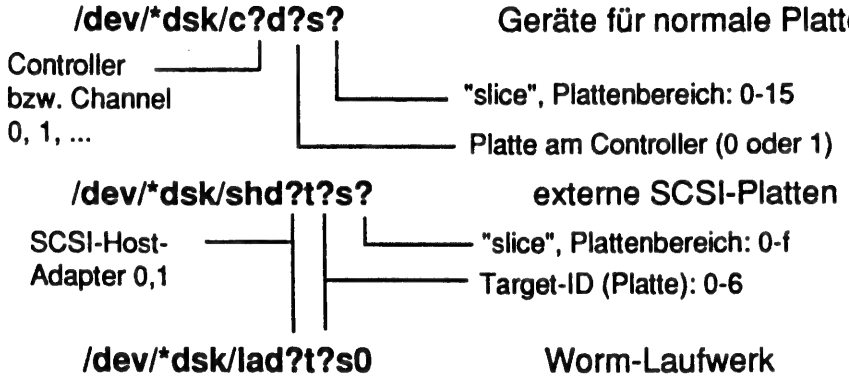
Gesamtgröße der "swap"-Bereiche:

SINIX V5.4 : **mindestens 2 mal Hauptspeicherausbau !**

Das Verzeichnis /dev

In SINIX/UNIX 5.4 sind viele Gerätedateien in Unterverzeichnissen von /dev zusammengefaßt:

/dev/dsk/	"block"-Geräte für Platten und Disketten
/dev/rdsk/	"raw"-Geräte für Platten und Disketten



/dev/VP/	Pseudo-Geräte für Spiegelplatten-System
/dev/rmt/c0s0	Streamer (rewind), rmt == "raw magn. tape"
c0s0n	Streamer (no rewind)
/dev/term/tty???	Terminals und Drucker SR-Baugruppe : tty000 - tty095 SIM-Baugruppe (TASCI) : tty200 - tty455 SX-Baugruppe (Modem) : tty500 - tty595
/dev/sim/	spezielle Geräte für TASCI
/dev/col/	Pseudo-Geräte für Grafik-COLLAGE
/dev/pts/	Pseudo-Geräte für Alpha-COLLAGE und LAN
/dev/dlpi/exos	Geräte für EXOS (LAN)
/dev/inet/	Geräte für Netzanschlüsse

Aus Kompatibilitätsgründen bestehen bei vielen Gerätedateien Verweise auf zusätzliche Namen:

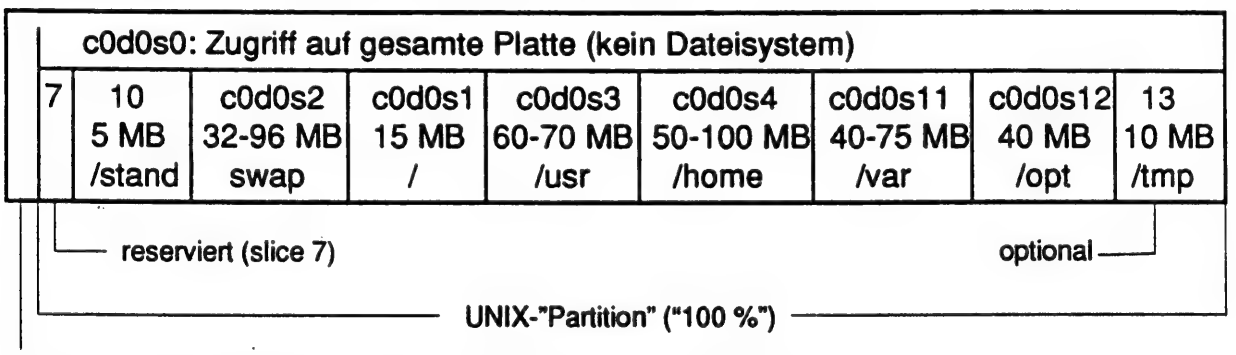
Beispiele: /dev/rmt/c0s0 === /dev/tape
/dev/pts/0 === /dev/Pts000

Gerätedateien für Plattenbereiche:

Bei den Standard-Platten können Sie an jeder Plattensteuerung (== **"Controller"**) 2 Platten anschließen.

Die Namen der Gerätedateien hängen davon ab, an welchem Channel bzw. Controller und an welchem Anschluß dieses Controllers die betreffende Platte angeschlossen ist.

Standardeinteilung bei einer "380 MB"-Platte:



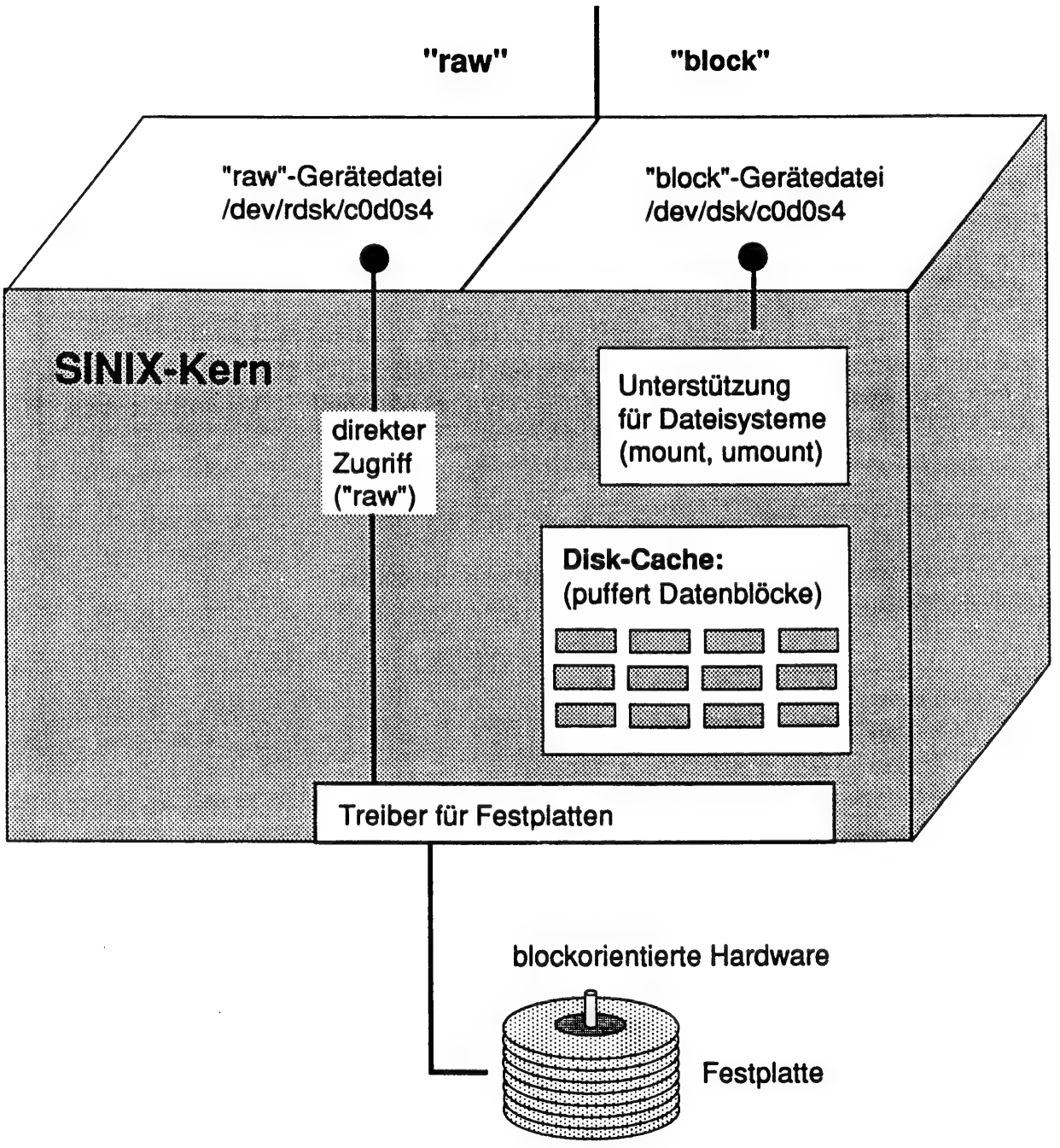
Zylinder 0: Platteneinteilung: "fdisk" und "VTOC" (virtual table of contens)
"boot"-Programm (Kopie von "/etc/boot")

Gerätedateien für 2. Platte:

- Ist die 2. Platte am gleichen Controller angeschlossen wie die 1. Platte, so heißen die Gerätedateien:

- Ist die 2. Platte an einem eigenen Controller angeschlossen, heißen die Gerätedateien:

"block" und "raw" Gerätedateien



"raw"-Gerätedateien sind **nicht** zeichenorientiert!

"block"- Gerätedateien:

Platten- und Diskettenlaufwerke sind blockorientierte Geräte, d.h. der SINIX-Systemkern kann nur ganze Datenblöcke lesen oder schreiben.

Der Systemkern puffert alle Daten, die über das Dateisystem oder über "block"-Gerätedateien gelesen oder geschrieben werden, in einem umfangreichen "Disk-Cache" (ca. 500 bis 1000 KB Arbeitsspeicher).

Die bisher besprochenen Gerätedateien für die Plattenbereiche sind "block"-Gerätedateien :

```
/dev/dsk/c0d0s1
/dev/dsk/c0d0s3
/dev/dsk/c0d0s4
```

"raw"-Gerätedateien:

Bei einigen Programmen oder Kommandos ist es sinnvoll, daß sie direkt auf die Platte zugreifen, ohne den "Disk-Cache" zu verwenden. Diese Programme/ Kommandos verwenden die "raw"-Gerätedateien:

```
/dev/rdsk/c0d0s1
/dev/rdsk/c0d0s3
/dev/rdsk/c0d0s4
```

Label `/dev/rdsk/c0d0s0`

Vorteil : schnelle Übertragung großer Datenmengen

Anwendung : - Turbo-Informix
- Kommandos: mkfs, fsck, dump
- Labelbereich lesen

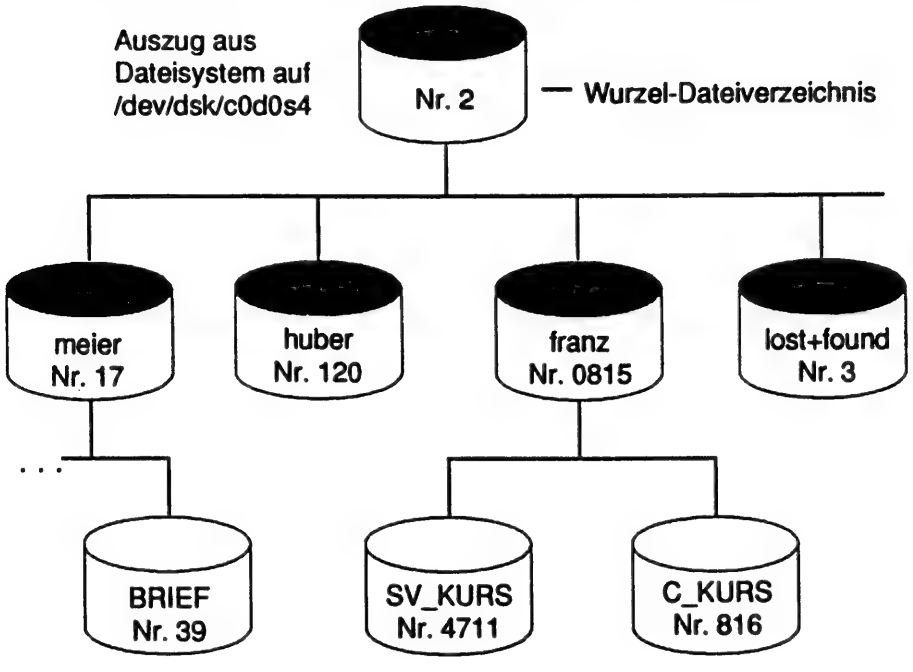
Achtung: "raw"-Gerätedateien sehen aus wie zeichenorientierte Gerätedateien, man kann jedoch nur in Blöcken zugreifen !!

nicht verwendbar für: mount, umount

Was ist ein Dateisystem ?

a) Aus Sicht des Systemverwalters:

Ein Dateisystem ist ein **eigenständiger Dateibaum,** der ein **oberstes Dateiverzeichnis** (Indexnummer 2), Unterverzeichnisse und Dateien enthält.



Jedes Dateiverzeichnis und jede Datei besitzt eine Verwaltungsnummer (Indexnummer), die innerhalb des Dateisystems eindeutig ist.

b) Aus Sicht des Systemkerns:

Ein Dateisystem besteht aus Datenstrukturen und Datenblöcken, die der Systemkern benötigt, um Dateiverzeichnisse und Dateien zu verwalten:

- Verwaltungsinformationen (**"Superblock"**)
- Inhaltsverzeichnis (**"inodes"/"Indexeinträge"**)
- Datenblöcke für Dateien und Dateiverzeichnisse

> Ein Dateisystem benötigt ca. 15% der formatierten Kapazität eines Plattenbereichs für Verwaltungsdaten bzw. für ein Minimum an freien Blöcken und Teilblöcken (Fragmenten).

Plattentypen und Plattendaten

Plattentypen beim MX300:

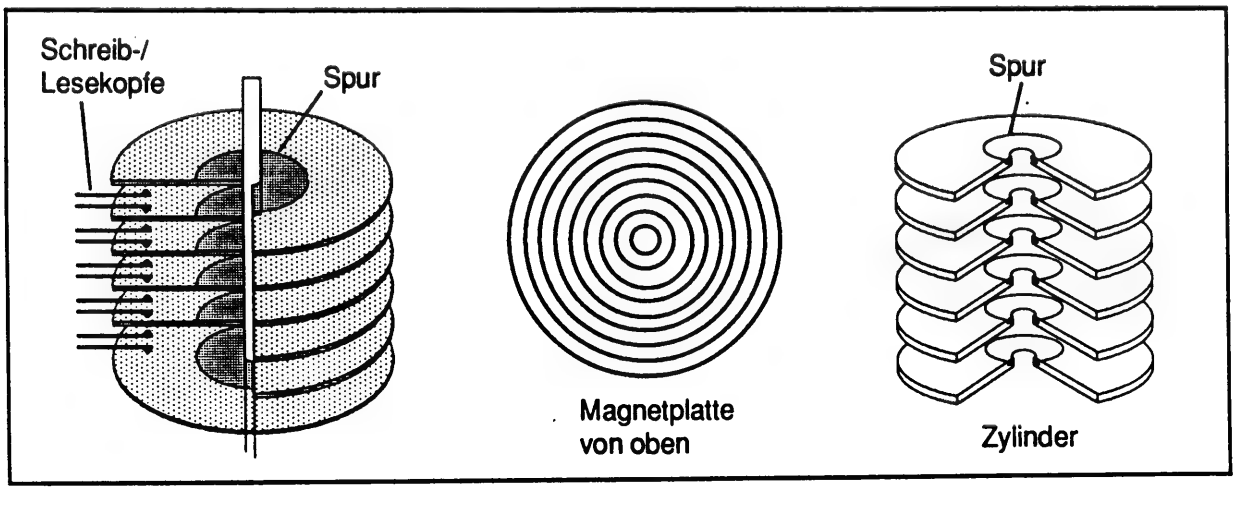
- "170 MB": MC1355, formatiert 139 MB (nur MX300-45)
- "300 MB": MegaFile/ME1300, formatiert 248 MB oder MC0558 (alt)
- "380 MB": MC1558, formatiert 310 MB oder MC1664, formatiert 322 MB
- "760 MB": MC1568 oder wren6, formatiert 649 MB oder ...

Plattendaten:

	"170 MB" MC1355	"380 MB" MC1558	"760 MB" MC1568	"380 MB" ... MC1664
Plattenköpfe: heads bzw. ntrack	8	15	15	7
Sektoren pro Spur —> nsect	34	34	53	53
ntrack*nsect == Sektoren/Zylinder	272	510	795	371
	(ein Sektor == 512 Bytes)			
==> KB / Zylinder	136	255	397	185

Plattendaten ausgeben lassen:

prtvtoc -p /dev/rdsk/c?d?s0



Standardeinteilung der Festplatten beim MX300:

Plattentyp/ Hsp.-Ausbau	MC1355 4-8 MB	MC1558 4-16 MB	MC1558 MB 32-64 MB	MC1664 4-16 MB	MC1664 32-64 MB	MC1568 8-16 MB	MC1568 32-64
Bruttokap.	170 MB	380 MB	380 MB	380 MB	380 MB	760 MB	760 MB
Nettokap.	136 MB	303	303 MB	322 MB	322 MB	632 MB	632 MB
Zylinder a*)	1017	1217	1217	1773	1773	1625	1625
/stand b*)	5(38)	5(21)	5(21)	5(28)	5(28)	5(12)	5(12)
/de swap b*)	16(122)	32(129)	96(387)	32(175)	96(530)	32(82)	128(330)
/ b*)	15(110)	15(60)	15(60)	15(28)	15(85)	15(38)	15(38)
/usr b*)	43(320)	60(241)	60(241)	60(330)	60(330)	60(155)	60(155)
/home b*)	27(207)	101(405)	37(147)	120(660)	55(305)	355(915)	252(649)
/var b*)	15(110)	40(160)	40(160)	40(220)	40(220)	75(192)	82(210)
/opt b*)	15(110)	40(160)	40(160)	40(220)	40(220)	70(180)	70(180)
/tmp b*)	------	10(419	10(41)	10(55)	10(55)	20(51)	20(51)

a*) : Es stehen effektiv 7 Zylinder mehr zur Verfügung, die jedoch reserviert sind.
b*) : Der erste Wert kennzeichnet die Slicegröße in MB (vor Generierung eines Filesystems).
Der Wert in Klammern stellt die korrespondierende Zylinderzahl des Slice dar.

System-Ressourcen beim MX300:

Verbrauchte Ressourcen nach der Installation

/Plattentyp	MC1355 a*)	MC1558 b*)	MC1568 b*)
/stand d*)	1,5 MB	2,0 MB	2.0 MB
/ d*)	8,5 MB	10,0 MB	10,0 MB
/usr d*)	36,0 MB	47,5 MB	47,0 MB
/home d*)	0,0 MB	0,0 MB	0,0 MB
/var d*)	1,0 MB	1,5 MB	1,5 MB
/opt d*)	1,5 MB	1,5 MB	1,5 MB
/tmp d*)	im / Bereich	0,0 MB	0,0 MB

Achtung:

- Der /usr-Bereich muß meist größere als 60 MB sein! Empfehlenswert sind 70 - 80 MB.
- Der /opt-Bereich sollte je nach Umfang der Anwendersoftware 40 - 80 MB großsein.

Dateisystemtypen

Bei der Installation des Betriebssystems können Sie für jeden "slice" (außer 0 und 7) festlegen:

- die Größe in Zylindern
- den Dateisystemtyp: "s5", "ufs"

  /stand hat immer den Dateisystemtyp: "bfs" !

Dateisystemtyp "bfs" (für /stand):

⇨ Primitives **Boot File System** für den Systemkern.

Dateisystemtyp "s5"

- original AT&T System V Dateisystem
- ein Superblock, eine Inode-Liste
- Datenblockgröße wahlweise: 512 B, 1 KB oder 2 KB
- Dateinamen bis 14 Zeichen
- Datenblöcke einer Datei werden im ungünstigsten Fall über das gesamte Dateisystem verteilt:

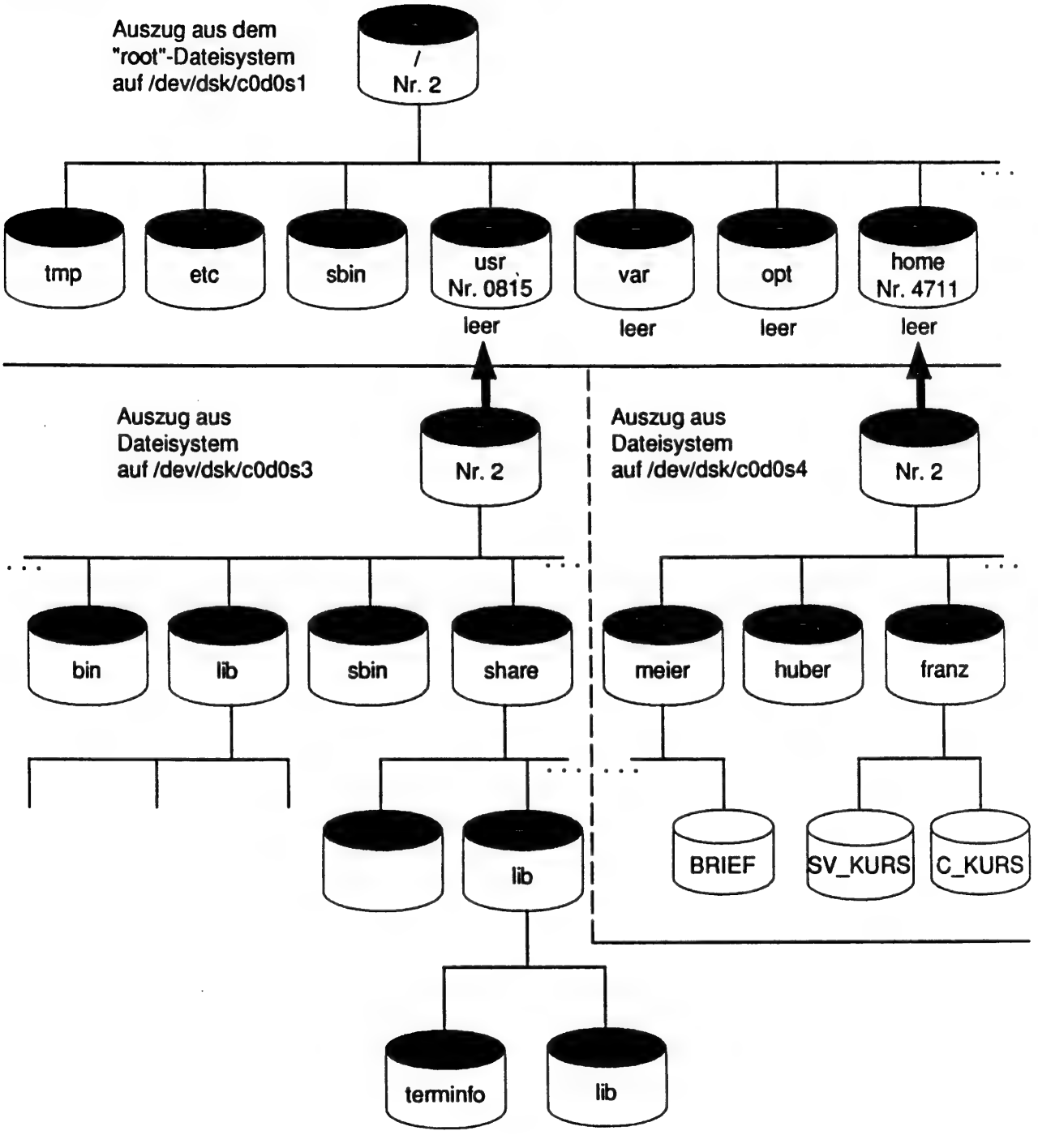
  * Schreiben ohne Optimierung (ohne CPU-Belastung)
  * langsames Lesen (Datenblöcke zusammensuchen)

- Dateisysteme sollten regelmäßig **reorganisiert** werden (Spezialkommando: "dcopy")
- eventuell notwendig zum Ablauf älterer UNIX-Software
- "s5"-Dateisysteme werden meist auf Disketten benutzt

Dateisystemtyp "ufs"

- **UNIX Fast File System** vom Berkeley-UNIX
- Dateisystem ist unterteilt in Zylindergruppen
- jede Zylindergruppe besitzt eine Kopie des Superblocks, eigene Verwaltungsinformationen, eigene Inodes und eigene Datenblöcke (—> Sicherheit)
- Datenblockgröße wahlweise: 4 KB oder 8 KB
- bei kleinen Dateien wird als letzter Datenblock ein "Fragment" zwischen 1 KB und 7 KB verwendet
- Dateinamen bis 255 Zeichen
- Datenblöcke einer Datei werden möglichst in einer Zylindergruppe angelegt:
  * Schreiben kostet Rechenzeit (wegen Optimierung)
  * Daten können schnell gelesen werden
- benötigt ca. 10% freien Platz, sonst ist keine Optimierung möglich
- es ist keine Reorganisation des Dateisystems notwendig
- durch größere Blockgröße erheblich schnellerer Zugriff auf große Dateien
- **sollte immer verwendet werden, falls alle Anwendungsprogramme das "ufs"-Format für Dateiverzeichnisse unterstützen**

Ausschnitt aus den Dateibäumen der Dateisysteme:



Hardware, Plattenbereiche und Dateisysteme

Platteneinteilung: /etc/vfstab

In die Datei /etc/vfstab müssen Sie Plattenbereiche und Dateiverzeichnisse eintragen, die beim Hochfahren automatisch überprüft und montiert werden sollen.
Außerdem stehen Informationen zum Montieren von Installationsdisketten dieser Datei (siehe "Installation ...").

Swap-Bereiche werden <u>nicht</u> in diese Datei eingetragen! "/dev/dsk/c0d0s2" wird automatisch als "swap"-Bereich genutzt, weitere "swap"-Bereiche müssen Sie mit dem Kommando "swap" hinzufügen.

Beispiel: **MX300-50 mit 2 MC1558 Platten:**

```
/dev/root          /dev/rroot          /        ufs   1   yes   -
/dev/dsk/c0d0s3    /dev/rdsk/c0d0s3    /usr     ufs   1   yes   -
/dev/dsk/c0d0s4    /dev/rdsk/c0d0s4    /home1   ufs   1   yes   -
/dev/dsk/c0d0s10   /dev/rdsk/c0d0s10   /stand   bfs   1   yes   -
/dev/dsk/c0d0s11   /dev/rdsk/c0d0s11   /var     ufs   1   yes   -
/dev/dsk/c0d0s12   /dev/rdsk/c0d0s12   /opt     ufs   1   yes   -
/dev/dsk/c1d0s1    /dev/rdsk/c1d0s1    /tmp     ufs   1   yes   -
/dev/dsk/c1d0s3    /dev/rdsk/c1d0s3    /home    ufs   1   yes   -
/dev/dsk/c1d0s4    /dev/rdsk/c1d0s4    /home2   ufs   1   yes   -
/proc                  -               /proc    proc  -   no    -
/dev/fd                -               /dev/fd  fdfs  -   no    -
# fuer Montieren von Installationsdisketten; Kommando: pkgadd
/dev/dsk/f0t       /dev/rdsk/f0t       /install s5    -   no    nosuid
/dev/dsk/f1t       /dev/rdsk/f1t       /install s5    -   no    nosuid
/dev/dsk/f0        /dev/rdsk/f0        /install s5    -   no    nosuid
/dev/dsk/f1        /dev/rdsk/f1        /install s5    -   no    nosuid
```

mount-Gerät ("block") | fsck-Gerät ("raw") | Montierverzeichnis | Typ des Dateisystems | fsck-pass | automatisch montieren | Optionen für mount, mit Komma getrennt

Dateisystemtyp:

s5	original UNIX System V
ufs	Berkeley "fast file system"
bfs	boot file system (nur /stand)
proc	Pseudo-Dateisystem zum Zugriff auf Prozeßdaten
fdfs	Pseudo-Dateisystem zum Zugriff auf File-Descriptoren
nfs	network file system (Berkeley)
rfs	remote file sharing (AT&T)

Kommandos für Dateisysteme

Systemverwalterkommandos finden Sie in folgenden Dateiverzeichnissen:

/sbin/
/usr/sbin/
/usr/ucb/
/etc/fs/.../

mkfs neues Dateisystem einrichten

Höchst gefährliches Kommando !
Nur auf freie Plattenbereiche anwenden !
Eventuell vorhandene Daten werden überschrieben !

```
mkfs -F ufs /dev/rdsk/c0d1s1 163200 34 15
```

mount Dateisystem montieren/einhängen

Abfragen, welche Dateisysteme wo montiert sind:

```
mount
```

Dateisystem an leeres Verzeichnis montieren:

```
mkdir DVZ
mount -F ufs /dev/dsk/c0d1s1 $HOME/DVZ
```

umount Dateisystem abmontieren

fsck -y Dateisystem überprüfen und bei Bedarf reparieren

```
fsck -F ufs -y /dev/rdsk/c0d1s1
```

Achtung: Sie dürfen **"fsck -y"** nur auf abmontierte Dateisysteme anwenden. Andernfalls wird das Dateisystem nicht repariert, sondern beschädigt !

df -k Plattenbelegung überprüfen

fsck, mount, umount

Unterschiedliche Programme für die Dateisystemtypen

/etc/fs/s5	df
	fsck
	fsdb
	labelit
	mkfs
	mount
/etc/fs/ufs	df
	fsck
	labelit
	mkfs
	mount
/etc/fs/bfs	mkfs
	mount
/usr/lib/fs/nfs	mount
	umount

Gemeinsame Kommandos unter /sbin

Sie rufen normalerweise ein Kommando unter "/sbin" auf und übergeben den Typ des Dateisystems mit dem Schalter "-F FStype" (FStype: "s5", "ufs" oder "bfs"):

```
df [-F FStype]
fsck [-F FStype] ...
labelit [-F FStype] ...
mkfs [-F FStype] ...
mount [-F FStype] ...
umount
```

Falls Sie den Dateisystemtyp nicht angeben, sucht das Kommando die Zeile in der "/etc/vfstab".
Ist kein Eintrag in der "vfstab" vorhanden, so setzt es den Typ auf "s5".
Anschließend ruft das Kommando das entsprechende Programm auf unter /etc/fs/.. oder /usr/lib/fs/.. .

mount, umount

mount -F ufs /dev/dsk/c1d0s4 /home1

Ist das Dateisystem in die Datei "/etc/vfstab" eingetragen, genügt beim "mount"-Kommando:

mount /home1

umount /home1

oder

umount /dev/dsk/c1d0s4

"umount" kann nur erfolgreich sein, wenn kein Prozeß auf das Dateisystem zugreift und wenn das Dateisystem keinen Montierpunkt eines anderen Dateisystems enthält (Reihenfolge !!!).

Falls "umount" nicht funktioniert:

```
fuser -u /dev/dsk/c1d0s4      # nachsehen, welche Prozesse
                              # auf /home1 zugreifen
```

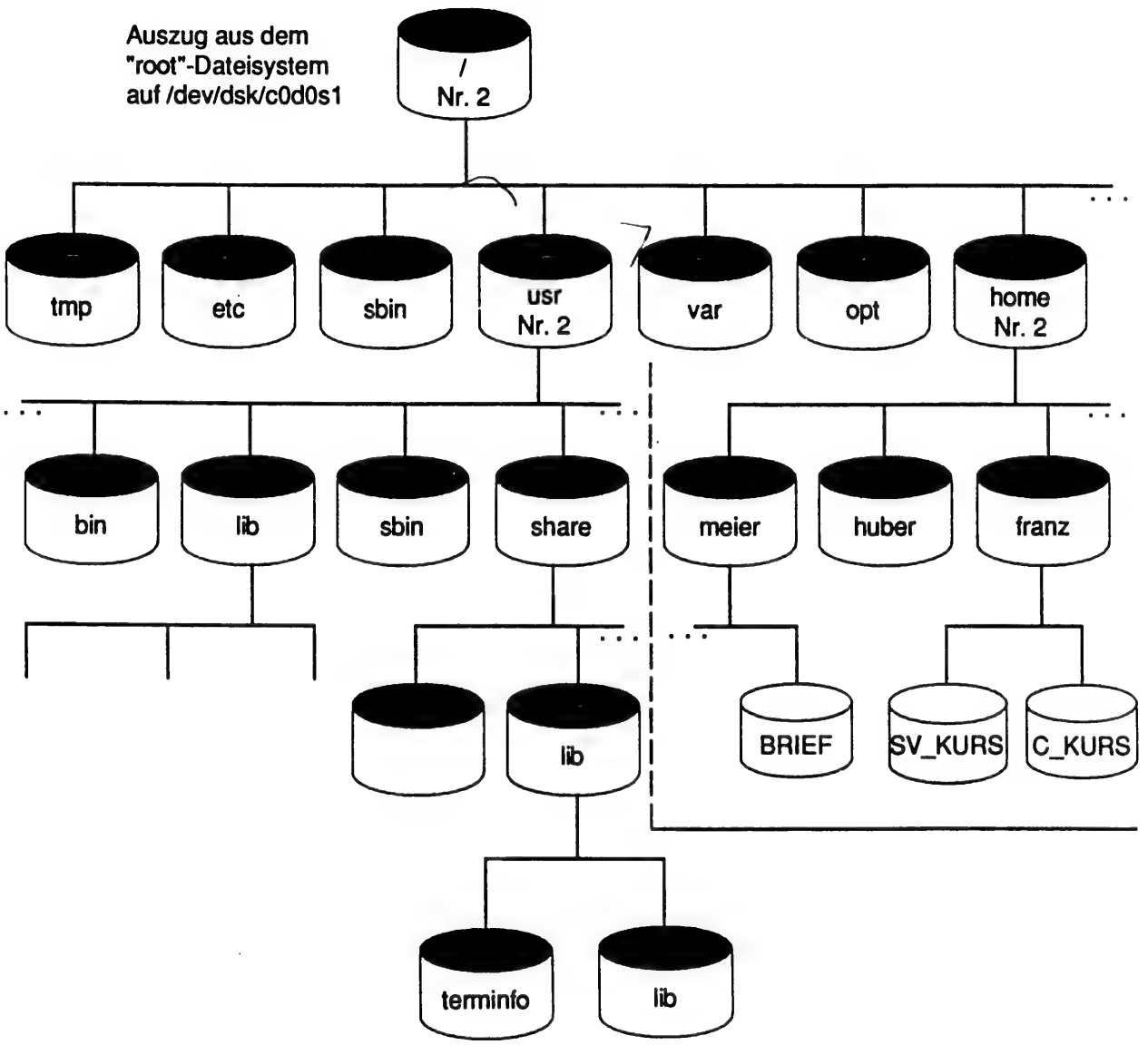
umountall

Die Prozedur **"umountall"** versucht, alle "ufs", "s5", "rfs" und "nfs" Dateisysteme abzumontieren, die in der "/etc/mnttab" eingetragen sind.
Die Dateisysteme **"/proc", "/stand"** und **"/dev/fd"** werden **nicht** abmontiert !

Datei: /etc/mnttab"

Das "mount"-Kommando trägt jedes montierte Dateisystem in die Datei "/etc/mnttab" ein und "umount" löscht jedes abmontierte Dateisystem aus der "/etc/mnttab". Viele Kommandos lesen aus der "/etc/mnttab".

Dateibäume nach "mount"



Dateisysteme neu erzeugen ?

Normalerweise rufen Sie in SINIX V5.4 Hilfsprogramme (z.B. "disksetup") und Prozeduren auf, die für Sie das "mkfs"-Kommando starten.

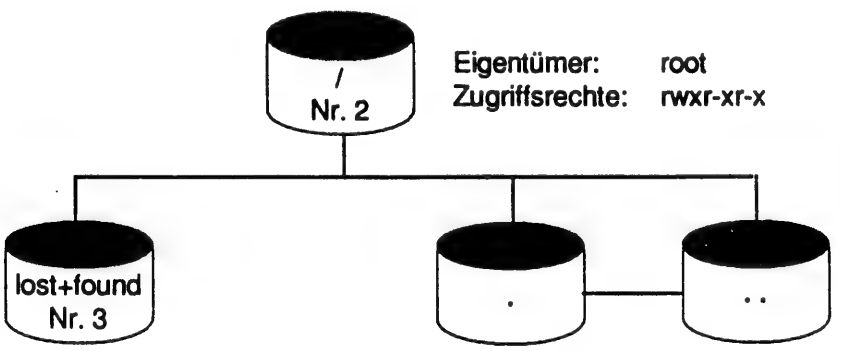
Was bewirkt "mkfs" ?

"mkfs" erzeugt im angegebenen Plattenbereich bzw. "slice" ein neues Dateisystem:

- **Eventuell vorhandene Daten werden dabei überschrieben!**
- wenn Sie "mkfs" als "root" aufrufen haben Sie noch 10 Sekunden Zeit, "mkfs" mit DEL abzubrechen

Im neuen Dateisystem richtet **"mkfs"** das oberste Dateiverzeichnis mit den Einträgen "." und ".." und ein Unterverzeichnis **"lost+found"** ein.

⇨ Mini-Dateibaum:



Beispiel für "ufs"-Dateisystem im Bereich "c1d0s4":

Vorbereitungen:

- eventuell auf dem "slice" vorhandene Daten sichern
- die Größe (length) des "slice" feststellen
- "slice" abmontieren: **umount** /dev/dsk/c1d0s4

```
mkfs -F ufs /dev/rdsk/c1d0s4 425850 34 15
#                            length nsect ntrack
```

mkfs-Kommando

Wenn Sie "mkfs" anrufen, müssen Sie immer die Größe (length) des slice angeben (in 512-Blöcken). Zusätzlich sollten Sie auch Plattenparameter angeben, damit "mkfs" Zylindergruppen in der optimalen Größe anlegt.

"mkfs"-Parameter bei "ufs":

nsect	Anzahl Sektoren pro Spur		
ntrack	Anzahl Köpfe bzw. Spuren pro Zylinder		
bsize	Dateisystemblockgröße:	8192 oder 4096	
fragsize	Fragmentgröße:	1024	
cgsize	Zylinder pro Zylindergruppe:	Standard	16
free	min. freier Platz in Prozent:	Standard	10
rps	Umdrehungen der Platte/Sek:	Standard	60
nbpi	Anzahl Datenbytes/Inode:	Standard	2048

Parameterangabe für "mkfs" über den Schalter "-o"

```
mkfs -F ufs -o nsect=34,ntrack=15,cgsize=8  /dev/rdsk/c1d0s4 425850
                                                              length
```

Parameterangabe für "mkfs" über eine Liste

```
mkfs -F ufs  /dev/rdsk/c1d0s4 45850    34    15    8192  1024   8 ...
#                             length nsec  ntrack  bsize fragsize cgsize ...
#                    unbedingt die Reihenfolge der Werte beachten!
```

Die Größe **"length"** müssen Sie immer angeben!

Parameter eines existierenden Dateisystems ausgeben:

```
mkfs  -F  ufs  -m  /dev/rdsk/c0d0s13
```

Anzahl der "inodes"

Die Anzahl der "inodes", die "mkfs" anlegt, begrenzt die maximale Anzahl von Dateien und Verzeichnissen, die Sie später anlegen können

Anzahl "inodes" überprüfen:

```
df -F ufs -o i /dev/rdsk/c1d0s4
```

Wie viele "inodes" legt "mkfs" an?

Bei "ufs"-Dateisystemen gibt es 2 Regeln:

1) **Parameter: nbpi** Anzahl Datenbytes, für die "mkfs" eine "inode" einrichtet

   Standardformel: ANZAHL = (length *512) / nbpi
   Standardwerte: nbpi = 2048 (erlaubt: n*1024)

2) **Pro Zylindergruppe max. 2048 "inodes"**

   Sind 2048 "inodes" pro Zylindergruppe erreicht, wirkt der Parameter "nbpi" nicht mehr.

"mkfs"-Parameter bei "s5"

```
mkfs -F s5 [-b bsize] /dev/rdsk/... length [:inodes] [gap blocks/cyl.]
```

"length" (in sektoren von 512 Bytes) müssen Sie angeben.
"bsize" (Dateisystemblockgröße: Standard 1024), die Anzahl der Inodes und Werte für "gap" und "blocks/cylinder" können Sie angeben:

Dateisystem auf Diskette:

```
mkfs -F s5 /dev/rdsk/f03ht 2844:711 2 36 # Diskette
```

"s5"-Dateisystem auf Platte:

```
mkfs -F s5 /dev/rdsk/c0d0s13 62220 7 510 #Platte
```

Parameter eines existierenden Dateisystems ausgeben:

```
mkfs -F s5 -m /dev/rdsk/f03ht
```

Aufgabe 3

Der Kursleiter teilt Ihrer Arbeitsgruppe einen freien Plattenbereich zu, den Sie für die Aufgaben verwenden können.

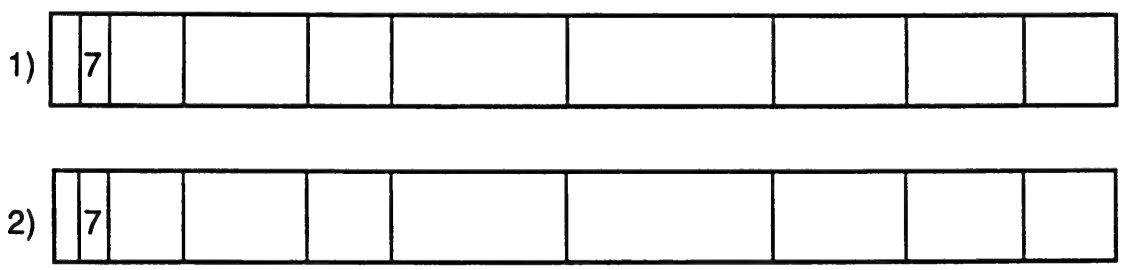
Im Dateiverzeichnis "/home/sv0/sbin" stehen für Sie spezielle Kommandos für "mount", "umount" und "chown" zur Verfügung, die Ihnen gezielt "Super-User" -Rechte verleihen.

a) Erstellen Sie sich folgende ".profile"-Datei:

```
PATH=/home/sv0/sbin:$PATH:/sbin:/usr/sbin:
LANG=De_DE.646
export PATH LANG
```

Melden Sie sich anschließend neu an. Rufen Sie kein Kommando mit dem absoluten Pfadnamen auf (nur "mount", nicht "/sbin/mount").

b) Stellen Sie mit "prtvtoc" und "df" die Platteneinteilung für beide Platten des Kursrechners fest.
Wie groß sind die einzelnen "slices"?
Wie heißen die Gerätedateien ?



c) Erzeugen Sie mit "mkfs" in ihrem Plattenbereich ein neues leeres "ufs"-Dateisystem und prüfen Sie es mit "fsck".

d) Erzeugen Sie ein Datei**verzeichnis** in Ihrem Login-Verzeichnis und montieren Sie Ihr Datei**system** mit "mount" an dieses Datei**verzeichnis.**
Überprüfen Sie Indexnummer, Zugriffsberechtigungen und Eigentümer Ihres Verzeichnisses vor und nach dem "mount"-Kommando ("ls -ild DVZ").

e) Ändern Sie den Eigentümer mit dem speziellen "chown".

f) Wechseln Sie in Ihr Dateiverzeichnis und kopieren Sie die /etc/passwd und Ihre "$HOME/.profile"-Datei in Ihr Dateisystem.
Prüfen Sie mit "ls -ial" die Indexnummern der neu angelegten Dateien und Dateiverzeichnisse.

g) Montieren Sie das Datei**system** mit "umount" wieder ab und überprüfen Sie, ob das Dateiverzeichnis leer ist. Unter welchen Umständen kann der "umount" ein Dateisystem nicht abmontieren ?

h) Überprüfen Sie Ihr Dateisystem mit "fsck".

i) Montieren Sie Ihr Dateisystem an ein **anderes** Dateiverzeichnis in Ihrem HOME-Verzeichnis und prüfen Sie den Inhalt.

j) Montieren Sie das Dateisystem mit "umount" wieder ab und geben Sie folgende Kommandos ein:

```
cd $HOME
echo "Dies ist mein HOME-Verzeichnis" > datei1
ls -il datei1
cd ..
mount -F ufs /dev/????  $HOME
                  |
                  └──── Ihre Gerätedatei
cd $HOME
ls -ial
```

Erklären Sie das Ergebnis! Wo ist die "datei1" ?

Was passiert, wenn Sie sich ab- und wieder anmelden ?
Was müssen Sie tun, um Ihre "datei1" wieder zu erhalten ?

Zusatzaufgabe:

Erzeugen Sie ein "s5"-Dateisystem und widerholen Sie d) bis g)

Zusatzaufgabe: symbolische Verweise

k) Überlegen Sie sich alle Unterschiede zwischen normalen Verweisen ("**Links**", "**ln**"-Kommando) und symbolischen Verweisen ("**ln -s**"). Probieren Sie dazu folgende Kommando-Folgen aus und erklären Sie die Ergebnisse:

Kommadofolge für normalen Verweis:

```
echo Hallo > dat1          # Datei erzeugen
ln dat1 dat2               # Verweis auf die gleiche Datei
ls -il dat?                # Indexnummern und Verweiszähler
                           # überprüfen
echo Aenderung >> dat1     # dat1 ändern
cat dat2                   # dat2 ansehen

rm dat1                    # einen Verweis löschen
cat dat2                   # Datei ist noch da

ln dat2 /tmp               # warum funktioniert das nicht ??
```

Kommandofolge für symbolischen Verweis:

```
echo Hallo2 > orig          # Datei "orig" erzeugen
ln -s orig verw             # symbolischer Link "verw"
ls -il orig verw            # Indexnummern und Verweis-
                            # zähler überprüfen

echo Aenderung >> orig      # Datei ändern
cat verw                    # Verweis ansehen
DVZ=`pwd`
ln -s $DVZ/orig /tmp/verw   # anderes Dateisystem

rm orig                     # Originaldatei löschen
ls verw                     # ist die Datei noch da ?
cat verw                    # eigenartige Fehlermeldung

cd /etc                        # A c h t u n g
ln -s passwd $HOME/testdat     # f a l s c h ! ! !
cd $HOME
cat testdat                 # wo liegt der Fehler ?
```

Zusätzliche Platte konfigurieren

Das Betriebssystem wird bei MX300-Rechnern bereits im Werk **auf der 1. Platte** vorinstalliert.
Um eine zusätzliche Platte zu aktivieren, müssen Sie:

- die UNIX Partition einrichten ("/sbin/fdisk")
- die Platteneinteilung und Dateisystemtypen festlegen ("/sbin/disksetup -I")
- /etc/partitions aktualisieren: "prtvtoc -e"

> **MX500-90:**
>
> Haben Sie in einen MX500-90 eine neue Platte eingebaut, so müssen Sie das Kommando **"diskadd"** aufrufen, um die Gerätedateien anzulegen. "diskadd" ruft dann automatisch "fdisk" und "disksetup" auf.

Beispiel für 2. Platte am 2. Controller:

```
fdisk    /dev/rdsk/c1d0s0
```

Sollte die Liste der aktiven Partitionen nicht leer sein, so sollten Sie zunächst die aktive Partition **löschen** (Auswahl 3).
Anschließend wählen Sie die Auswahl 1 und erzeugen **eine** neue UNIX Partition, die 100% der Platte belegt. Die UNIX-Partition muß aktiv sein.

```
disksetup -I     /dev/rdsk/c1d0s0
```

"disksetup" stellt Ihnen verschiedene Fragen:

- Anzahl der von Ihnen gewünschten "slices"
- absolute Pfadnamen der Dateisysteme (z.B.: /home1)
- gewünschter Dateisystemtyp: s5, ufs, na (no access) "na" ist z.B. sinnvoll für einen 2. "swap"-Bereich oder bei "raw"-Bereichen für Datenbanken
- **Größe der "slices" in Zylindern oder in Sektoren (je nach SINIX-Version)**

Eine genauere Beschreibung finden Sie in der SINIX-Freigabemitteilung.

disksetup

Platteneinteilung und Dateisystemtypen festlegen:
(vgl. vorhergehende Seite)

Sie können dem "**disksetup**"-Programm eine Datei angeben, in der Sie Standardwerte für die Platteneinteilung hinterlegt haben:

```
disksetup -d stdwert.dat -I /dev/rdsk/c1d0s0
```

Aufbau einer Datei mit Standardwerten für "disksetup":

```
 1     /roots     ufs     15M     #   ca. 15 MB
 2     swap2      -       2m      #   ca. 2 x Hauptspeicher
 3     /var_n     ufs     40M     #   ca. 40 MB
 4     /home1     ufs     60W     #   ca. 60 % vom Rest
 5     /home2     ufs     40W     #   ca. 40 % vom Rest
 6     /tmp_n     ufs     10M     #   ca. 10 MB
10     /stands    bfs     5M      #   ca. 5 MB
```

Dateisystemtyp oder "-"

slice mount-DVZ

In einer Standardwertdatei für disksetup sollte mindestens eine Zeile mit der Größenangabe "W" (Prozent vom Rest) vorkommen.

Was macht "disksetup" ?

- Platteneinteilung auf die Platte schreiben (VTOC)
- Dateisysteme mit "**mkfs**" einrichten
- mount-Dateiverzeichnisse anlegen
- "**/etc/vfstab**" erweitern
- Dateisysteme montieren

"disksetup" aktualisiert jedoch nicht die Datei "/etc/partitions" !

Sie müssen daher folgende Kommandos eingeben:

```
mv /etc/partitions /etc/partitions.old
prtvtoc -e /dev/rdsk/c0d0s0
prtvtoc -e /dev/rdsk/c0d1s0      # oder:
prtvtoc -e /dev/rdsk/c1d0s0      # (alternativ)
```

Platteneinteilung nachträglich ändern

In SINIX V5.4/UNIX V.4 können Sie die Platteneinteilung auch nachträglich ändern, z.B.:

- einen slice in mehrere slices aufteilen
- zwei slices zu einem slice zusammenfassen

Solche Änderungen erfordern große Vorsicht, da Sie den Startsektor und die Länge (size, Anzahl Sektoren) der slices selbst berechnen müssen. Die Größe eines "slices" sollte immer ein Vielfaches der Zylindergröße sein !!!

Sie haben 2 Kommandos zur Verfügung:

mkpart mit Schalter "-p" und "-P"
"mkpart" verwendet die Datei "/etc/partitions"

edvtoc -f
"edvtoc" verwendet eine Datei im Format, das Sie mit "prtvtoc -f .... " erzeugen können.

Beispiel für "edvtoc":

Die slices 4 und 5 (/home1 und /home2) aus dem letzten Beispiel zu einem slice zusammenfassen:

1. Schritt: Daten sichern, /home1 und /home2 abmontieren

2. Schritt: `prtvtoc -f vtoc1.dat /dev/rdsk/c1d0s0`

3. Schritt: Sie können nun die Datei "vtoc1.dat" editieren und die neue Platteneinteilung eintragen. Das Format der Datei "vtoc1.dat" finden Sie auf der nächsten Seite.

4. Schritt: neue Platteneinteilung in VTOC schreiben:

   `edvtoc -f vtoc1.dat /dev/rdsk/c1d0s0`

5. Schritt: neues Dateisystem mit "mkfs" erzeugen, "/etc/vfstab" modifizieren

Dateiformat: "edvtoc -f" und "prtvtoc -f"

Ausgabedatei von "prtvtoc -f vtoc1.dat /dev/...."
vtoc1.dat:

#SLICE	TAG	FLAGS	START	SIZE
0	0x5	0x201	510	621180
1	0x4	0x200	544	31110
2	0x4	0x201	31654	49470
3	0x4	0x200	81124	82110
4	0xb	0x200	163234	255510
5	0x4	0x200	418744	170340
6	0x4	0x200	589084	20910
7	0x1	0x201	510	34
8	0x0	0x0	0	0
9	0x0	0x0	0	0
10	0x4	0x200	609994	11696
11	0x0	0x0	0	0

Die Bedeutung der Spalten "TAG" und "FLAGS" finden Sie in der Datei "/usr/include/sys/vtoc.h":

TAG:

0x01	Boot slice
0x02	Root filesystem
0x03	Swap filesystem
0x04	Usr filesystem
0x05	full disk
0x09	Stand slice
0x0a	Var slice
0x0b	Home slice
0x0c	dump slice

FLAGS:

0x01	Unmountable partition
0x10	Read only
0x100	Partition open (for driver use)
0x200	Partition is valid to use

Wie müssen Sie die Datei "vtoc1.dat" verändern, um die slices 4 und 5 in einem slice zusammenzufassen ?

#SLICE	TAG	FLAGS	START	SIZE
...				
...				

"swap"-Bereiche konfigurieren

"**slice**" 2 der 1. Platte ist immer "**swap**"-Bereich !

Haben Sie eine 2. Platte nachträglich installiert, ist es sinnvoll, einen 2. "**swap**"-Bereich auf der 2. Platte einzurichten.

Mit dem Kommando "**swap**" können Sie einen zusätzlichen "swap"-Bereich aktivieren, deaktivieren oder die Liste der aktiven "swap"-Bereiche anzeigen lassen:

```
swap -a /dev/dsk/..... 0 length # aktivieren
```

- Sie müssen die "**block**"-Gerätedatei und die "length" (in 512 Byte-Blöcken) eines freien "slices" oder eine normale Datei angeben. Falls Sie eine normale Datei angeben, muß sie bereits die Größe von "length"*512 besitzen.
- der Parameter 0 steht für den Startblock, ab dem der "slice" zum "paging" und "swapping" verwendet wird
- das Kommando muß bei jedem Systemstart neu gestartet werden

```
swap -d /dev/dsk/..... 0      # "swap" deaktivieren

swap -l                       # Liste ausgeben
```

Wie gehen Sie vor ?

- Wählen Sie einen freien "slice", der nicht für ein Dateisystem genutzt wird.

  Beispiel: **/dev/dsk/c1d0s2** (2. Platte am 2. Controller)

- Schreiben Sie eine rc-Prozedur "addswap", die das Kommando "**swap**" aufruft (—>/etc/init.d/):

  Wie Sie dies tun können, erfahren Sie im Kapitel "rc-Dateien".

- Beim **MX500-90** tragen Sie zusätzliche swap-Bereiche in die Datei "**/etc/swaptab**" ein.

Sinnvolle Plattenaufteilung

Wenn Sie nur eine Platte besitzen, ist die Plattenaufteilung im wesentlichen vorgegeben.

Haben Sie jedoch 2 Platten (möglichst jeweils mit eigener Plattensteuerung), dann hängt die Leistungsfähigkeit Ihres Rechners davon ab, ob Sie die Plattenzugriffe gleichmäßig auf beide Platten verteilen.

Sie sollten außerdem dafür sorgen, daß alle Dateisysteme maximal bis 85 % belegt sind, da Sie sonst Geschwindigkeitseinbußen in Kauf nehmen müssen.

Beispiel:

1. Platte:

	s7	c0d0s10 /stand	c0d0s2 swap	c0d0s1 /	c0d0s3 /usr	c0d0s4 /home1	c0d0s11 /var	c0d0s12 /opt

2. Platte mit eigener Plattensteuerung (Controller):

	s7	c1d0s1 /tmp	c1d0s2 swap	c1d0s3 /home	c1d0s4 /home2

reserviert

Es wäre besser, wenn auch "/var" auf der 2. Platte liegen würde !

/tmp wurde hier bei der Installation nicht angelegt, aber später auf der 2.Platte eingerichtet!

Verteilte Dateisysteme (NFS)

Falls Ihr Rechner mit anderen Rechnern über TCP/IP-LAN vernetzt ist, können Sie mit Hilfe von NFS ("Network File System") Datei**verzeichnisse** eines anderen Rechners an Ihren Dateibaum "montieren".

Die Kommandos **"mount"**, **"umount"** und die Datei **"/etc/vfstab"** erhalten in diesem Zusammenhang eine erweiterte Funktionalität:

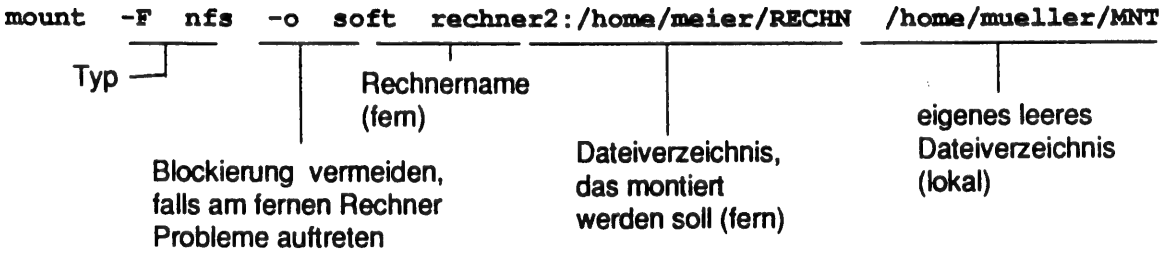
siehe: **mount(1M) (NFS)**

Datei**verzeichnisse** Ihres eigenen Rechners können Sie über das Kommando "share" für andere Rechner freigeben:

siehe Datei: **/etc/dfs/dfstab**

Datei**verzeichnis** eines anderen Rechners "montieren":

Das Dateiverzeichnis muß auf dem anderen Rechner in der Datei **"/etc/dfs/dfstab"** mit "share" für Ihren Rechner freigegeben sein.



Entsprechender Eintrag in der /etc/vfstab :

```
/dev/root /dev/rroot                    /         ufs      1    yes  -
/dev/dsk/c0d0s3 /dev/rdsk/c0d0s3        /usr      ufs      1    yes  -
/dev/dsk/c0d0s10 /dev/rdsk/c0d0s10      /stand    bfs      1    yes  -
/dev/dsk/c0d0s11 /dev/rdsk/c0d0s11      /var      ufs      1    yes  -
                                ....
/proc                           -       /proc          proc      -    no   -
/dev/fd                         -       /dev/fd        fdfs      -    no   -
rechner2:/home/meier/RECHN      -       /home/mueller/MNT  nfs  -  yes  rw,soft
```

mount-Gerät ("block") — fsck-Gerät ("raw") — Montierverzeichnis — Typ des Dateisystems — Optionen für mount

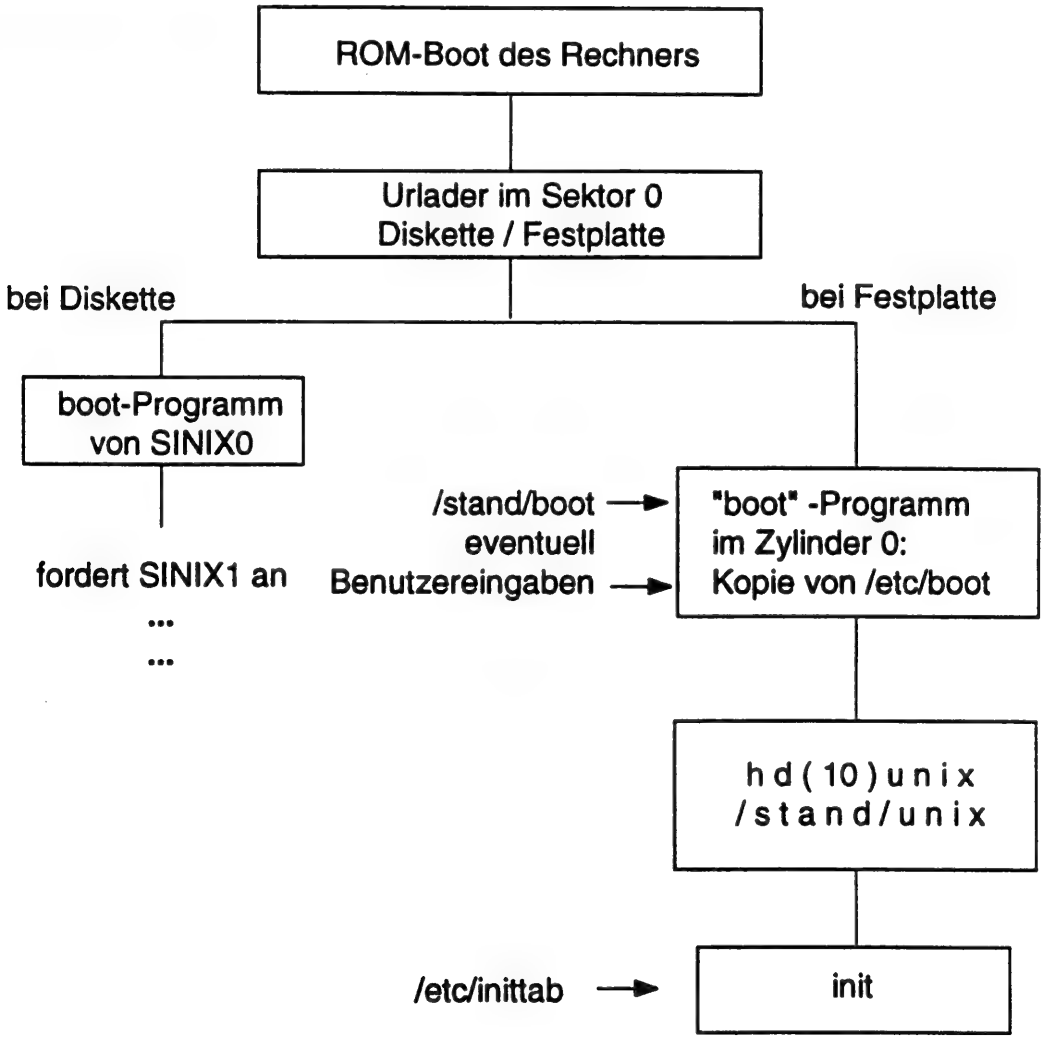
Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

Boot von Diskette oder Platte

Installationsdisketten (MX300):

SINIX0: Diskette mit "boot"-Programm
SINIX1: Miniroot-Dateisystem ("s5") mit Systemkern
SINIX2: Dateisystem ("s5") mit Kommandos und Installationsprozeduren

Systemstart beim MX300:



Systemstart beim MX500:

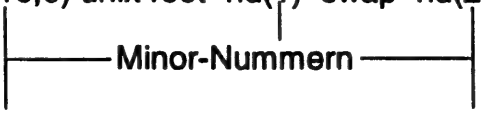
Beim MX500 bestimmen Sie über den Hardwaremonitor, ob der Rechner vom Band oder von der Platte bootet (siehe Kapitel 6 des Kursordners)

boot-Parameter

MX300: Datei /stand/boot (nicht /etc/default/boot)

AUTOBOOT = YES — bei "YES" wartet "boot" nicht auf
NO — Benutzereingaben sondern lädt sofort das Programm, das in DEFBOOTSTR angegeben ist:

DEFBOOTSTR=hd (10,0) unix root=hd(1) swap=hd(2)



slice 10 slice 1 slice 2

TIMEOUT=15 — falls AUTOBOOT=NO, wartet "boot" 15 Sekunden auf eine Benutzereingabe und lädt dann das Programm aus DEFBOOTSTR.

rootfstype=ufs — Typ des "root"-Dateisystems ("s5", "ufs")

Benutzereingaben für boot:

- im Fehlerfall: falls das Programm in DEFBOOTSTR nicht geladen werden kann

- bei AUTOBOOT=NO

Festplatte: —> hd(m,o) [pfadname]
oder hd(m) [pfadname]

Diskette: —> fd(m,o) [pfadname]
oder fd(m)[pfadname]

m — Minor-Device-Nummer des slices
o — Offset (kann weggelassen werden)

Beispiel: "booten" von Diskette mit "Ram-Disk":

fd(7)unix root=ramd(0) swap=ramd(1)

MX500-90 : siehe bootflags des Hardwaremonitors

Typischer Bootstring für MX500-90:
sp(0,10)unix root=ssd(m1) swap=ssd(m2) rootfstype=ufs
m1, m2 == Minornummern der slices

Systemstart, Einbenutzer-Betrieb, regelmäßige Aufgaben

Meldungen beim Systemstart

Alle Meldungen des SINIX-V5.4-Kerns finden Sie bei einem laufenden SINIX-System am Ende der Datei: **"/var/adm/klog.msg"**

- Meldungen beim Systemstart
- Fehlermeldungen im laufenden Betrieb

```
Tue Oct 30 09:36:23 GMT 1990
total real memory       = 12582912                    12 MB Arbeitsspeicher
total available memory  = 10629120
...

  device      address     iobase      ipl     maps[at]    comment
:Processor    -           -           -       -           i486: typ 00, frq 00, gs 01
:Multibus     FC000000    FD000000    -       FFF[000]
+exos0: WARNING: loopback test failed, check drop cable and tranceiver
+exos0 HW 0.0 NX 5.5, ether 08-00-14-15-50-56
:exos0        00001A00    00000000    01(5)   06F[000]
+hd0          -           -           -       -           drive 0: MC1558     1. Platte/1.Plattensteuerung
+fd0          -           -           -       -           drive 2: Floppy     1. Floppylaufwerk
+hd1          -           -           -       -           drive 0: MC1558     2. Platte/2.Plattensteuerung
:hd0          000073F8    00000000    02(5)   022[075]    Storager FW 34      1. Plattensteuerung
:hd1          000075F8    00000001    02(5)   022[09D]    Storager FW 34      2. Plattensteuerung
+it0          -           -           -       -           drive 0: TDC 3xxx
:is_ts0       000073FC    00000000    -       100[0BF]    Storager FW 34
:is_ts1       000075FC    00000001    -       100[1BF]
:sc0          E0000100    -           -       -
:sr0          00EF7000    00001000    -       -           fw 0x82, ... 2 x V.24, 4 x SS97
:sr1          00EF6000    00011100    -       -           fw 0x02, ... 6 x SS97
:sr2          00EF5000    00021200    -       -           fw 0x02, ...
#acc          -           -           -       -           unit(s) 8 SCSI-Treiber für "Juke-Box"
#exa          -           -           -       -           unit(s) 3 SCSI-Treiber für 2.0 GB Kassette
#fd           -           -           -       -           unit(s) 0
                          ...

#vp           -           -           -       -           unit(s) 0 Treiber für Spiegelplattensystem
Collage Line Discipline and Vtty Driver                   linecnt = 4
+exos0: WARNING: loopback test failed, check drop cable and tranceiver Ethernet-/LAN-Baugruppe
it0: error during tape operation,
iopb: A1 11 82 81 04 00 00 00 00 00 07 D0 10 0C 30 00 73 FC 00 00 00 00 00 00
fd0 on controller 0: HARD ERROR: filesystem blkno=2864, disk sector (79,1,2), scnt=16, code=0x29
iopb: 82 11 82 29 02 01 00 4F 00 02 00 10 10 07 5E AC 73 F8 00 00 00 00 00 00
```

└ Fehlermeldungen der Plattensteuerung

Feststellen von Hardwarefehlern

Wie erkennt man Hardwarefehler ?

Hardwarefehler jeglicher Art machen sich in jedem Fall durch Systemmeldungen **an der Konsole** bemerkbar. Diese Meldungen werden im Mehrbenutzerbetrieb protokolliert:

⇨ Datei: **"/var/adm/klog.msg"**

Ein harmloser **"Hardwarefehler"** liegt beispielsweise vor, wenn Sie auf eine schreibgeschützte Diskette oder auf ein leeres Diskettenlaufwerk schreiben wollen.

Die Systemmeldungen enthalten immer das Kürzel des betroffenen SINIX-Treibers plus der Nummer der Plattensteuerung, der Platte, der Diskette oder des Streamers.

Beispiele für MX300 (vgl. L 3- 1):

hd0 oder hd1	: 1. bzw. 2. Platten
fd0	: Diskettenlaufwerk
it0	: Streamer (Magnetbandkassette)
exos0	: 1. Ethernet Baugruppe (LAN)
exa ncr0	: Exabyte-Magnetbandkassettenlaufwerk

Im Anschluß an dieses Kürzel gibt der betroffene Treiber entweder eine eigene Meldung aus, oder er gibt die **"Fehlerbytes"** des Controllers in hexadezimaler Form weiter:

iopb : Fehlermeldung der Platten-/Disketten- und Streamer-Steuerung (Controller) in hexadezimaler Form
Diese hexadezimalen Ausgaben sind wichtig für den Wartungstechniker !

Aufgabe 4:

In Ihrem Dateiverzeichnis finden Sie die Datei **"klog.msg"**, die die Meldungen eines MX300-Kerns enthält.

Bitte stellen Sie den genauen Hardwareausbau dieses MX300 fest:

- Arbeitsspeicher
- Plattensteuerungen (Controller)
- Platten und Plattentypen, Diskettenlaufwerke
- E/A-Prozessoren
- DFÜ/LAN-Prozessoren

Wieviel Arbeitsspeicher benötigt der Systemkern ?

Von welcher Hardware stammen die Fehlermeldungen, die Sie in dieser Datei finden?

Zusatzaufgabe:

Wenn Sie sich die gesamte Datei genau ansehen, können Sie die Geschichte des betreffenden Systems verfolgen!

Arbeitsspeicherabzüge

Zu Diagnose-Zwecken sichert der Rechner nach folgenden Ereignissen den Inhalt des kompletten Arbeitsspeichers und Zusatzinformationen im Verzeichnis **"/var/crash"**.

- nach einem Systemabsturz oder DIAG+ RESET (sinnvoll)
- nach Stromausfall oder unsachgemäßem Abschalten des Rechners (Inhalt des Speichers leer)

Der Arbeitsspeicherabzug ist nur verwendbar, wenn der Rechner nach dem "Absturz" nicht ausgeschaltet wurde !!!

Der Rechner sichert den Arbeitsspeicher normalerweise in 2 Schritten:

1) physikalische Kopie in den "swap"-Bereich

Wurde das Betriebssystem nicht ordnungsgemäß heruntergefahren, kopiert der Rechner vor dem nächsten Systemstart den Inhalt des Arbeitsspeichers in den **"swap"**-Bereich der 1. Platte.

2) Sicherung nach "/var/crash":

Beim Systemstart wird beim Übergang in den Mehrbenutzer-Betrieb automatisch ein Programm **"savecore"** aufgerufen, das den Arbeitsspeicherabzug aus dem **"swap"**-Bereich (wo er überschrieben würde) nach **"/var/crash"** kopiert.

Der Arbeitsspeicherabzug hat die Größe des Arbeitsspeichers und belegt daher viel Platz auf der Platte.

Nach jedem Stromausfall oder nach unsachgemäßem Abschalten des Systems müssen Sie den Arbeitsspeicherabzug löschen.

"run-level", init

In SINIX V5.4 sind mehrere Betriebsarten (**"run-level"**) für das Betriebssystem definiert.
Für jede Betriebsart (jeden run-level) ist festgelegt, welche Prozesse in diesem run-level laufen dürfen:

run-level 0: Betriebssystem herunterfahren (keine Prozesse)

run-level 1: Einbenutzer-Betrieb: nur die Konsole wird bedient
Prozesse, die ohne Terminal oder an der Konsole laufen, werden jedoch nicht beendet ! Sie können solche Prozesse mit dem Kommando ***"killall 15"*** beenden.

run-level s,S: Einbenutzer-Betrieb mit Kennwortabfrage: die virtuelle Konsole **"syscon"** wird auf das Terminal gesetzt, an dem **"init s"** aufgerufen wurde. Hintergrundprozesse werden auch nicht beendet.

run-level 2: normaler Mehrbenutzer-Betrieb

run-level 3: Mehrbenutzer-Betrieb + NFS

run-level 6: herunterfahren und reboot

run-level a,b,c: zum Start zusätzlicher Anwendungen

run-level ändern *(Beispiele):*

a) `init s` # Wechsel in Einbenutzer-Betrieb
an beliebigem Terminal

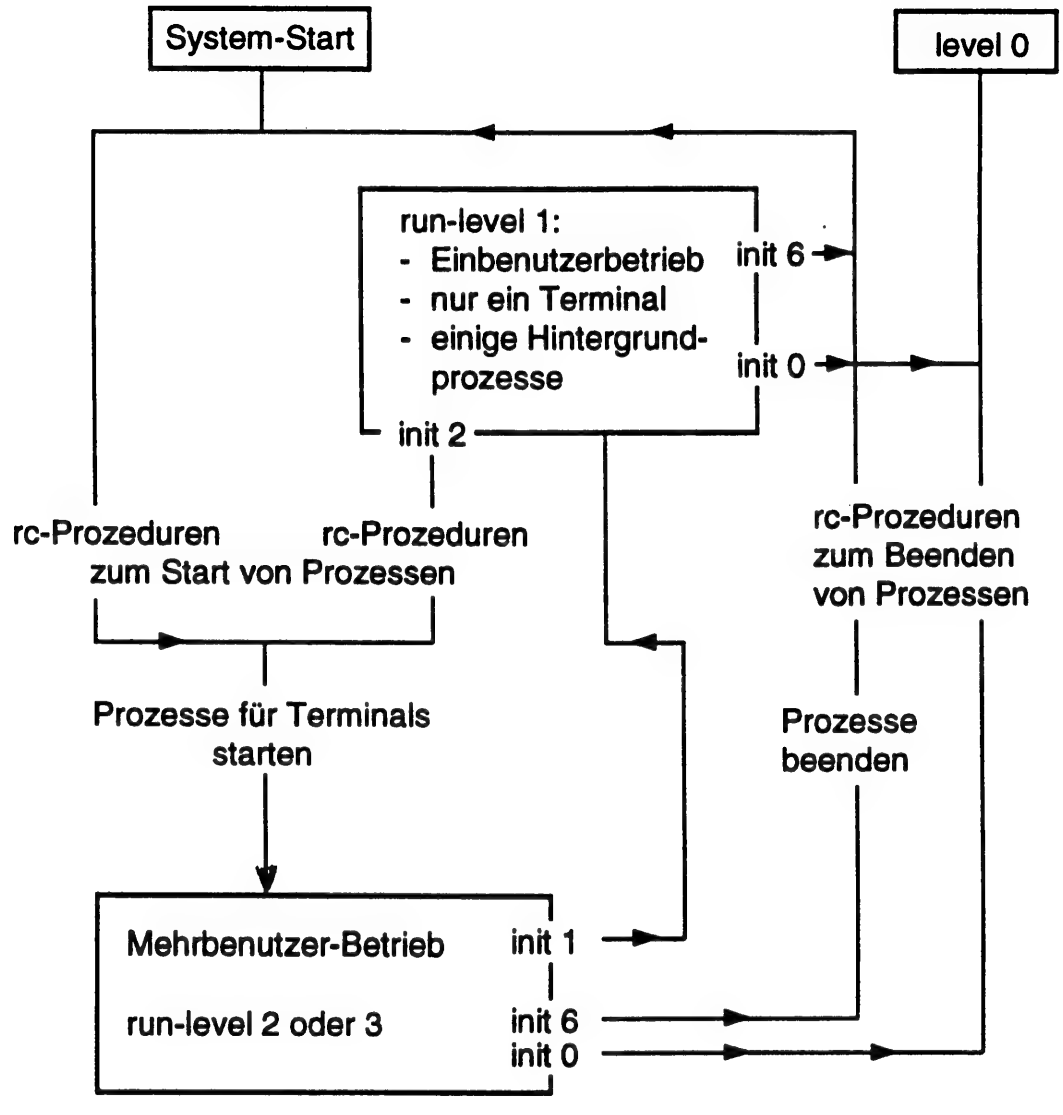
---

b) `sync; sync`
`init 0` # System schnell herunterfahren
(ohne Warnung der Benutzer)

Aktuellen run-level abfragen:

```
who -r
```

run level 0, 1, 2 und 3



/etc/inittab

In der "inittab" sind alle Prozeduren und Programme eingetragen, die der "init" beim Übergang in einen anderen run-level automatisch starten soll. Sie können zu jeder Prozedur/jedem Programm die **Aktion** des "init" festlegen:

respawn	Prozeß im Hintergrund starten: Falls sich der Prozeß beendet, Prozeß neu starten.
wait	Prozeß starten und Prozeßende abwarten
once	Prozeß einmal starten, ohne das Prozeßende abzuwarten
sysinit	Prozeß nur beim Systemstart starten
off	Prozeß (falls vorhanden) mit SIGTERM und SIGKILL beenden
initdefault	in dieser Zeile steht der run-level, in den das Betriebssystem hochfährt

Jede Zeile besitzt 4 Felder, getrennt durch Doppelpunkt:

```
bchk::sysinit:/sbin/bcheckrc </dev/console >/dev/sysmsg 2>&1
is:2:initdefault:
r0:0:wait:/sbin/rc0 off 1> /dev/sysmsg 2>&1 </dev/console
r1:1:wait:/sbin/rc1 1> /dev/sysmsg 2>&1 </dev/console
r2:23:wait:/sbin/rc2 1> /dev/sysmsg 2>&1 </dev/console
r3:3:wait:/sbin/rc3 1> /dev/sysmsg 2>&1 </dev/console
r5:5:wait:/sbin/rc0 reboot 1> /dev/sysmsg 2>&1 </dev/console
r6:6:wait:/sbin/rc6 reboot 1> /dev/sysmsg 2>&1 </dev/console
sd:0:wait:/sbin/uadmin 2 3 >/dev/sysmsg 2>&1 </dev/console
fw:5:wait:/sbin/uadmin 2 2 >/dev/sysmsg 2>&1 </dev/console
rb:6:wait:/sbin/uadmin 2 1 >/dev/sysmsg 2>&1 </dev/console
sc:234:respawn:/usr/lib/saf/sac -t 300 ,
sc1:23:off:/sbin/getty ttyc1 S
s000:12345:respawn:/sbin/getty term/tty000 S
s001:23:respawn:/sbin/getty term/tty001 S
s003:23:respawn:/sbin/getty term/tty003 S
```

Aktion — Shell-Kommando + Parameter — Verweis auf /etc/gettydefs

Liste der run-level oder leer

eindeutiger Identifikator pro Zeile (max. 4 Zeichen)

inittab verändern

Bei einer Neukonfiguration des Systemkerns (Kernel-Environment) wird die "/etc/inittab" eventuell neu erzeugt. Die "/etc/inittab" entsteht dabei aus folgenden Dateien:

/etc/conf/cf.d/init.base # Basis der inittab

/etc/conf/init.d/sc # Service-Konsole
/etc/conf/init.d/sr # Terminals
/etc/conf/init.d/... # ev. weitere Dateien

Eine dauerhafte Veränderung der Konfiguration sollten Sie daher auch in diesen Dateien vornehmen.

Betriebssystem herunterfahren

Die Rechner MX300 und MX500 müssen Sie unbedingt softwaregesteuert unter der Benutzerkennung **"root"** mit der **"shutdown"**-Prozedur oder mit **"init 0"** herunterfahren. Der MX300 schaltet sich anschließend automatisch ab.

Sie dürfen den MX300 niemals über die Ausschalttaste abschalten !!

Die Prozedur **"shutdown"**:

shutdown [-y] [-gmm] [-i[0sS156]]

-y — keine weiteren Abfragen

-g[0-9]* — Zeit bis zum shutdown (Standard: 60 Sek.)

-i[0sS156] — neuer run-level (Standard: 0)

Um die **"shutdown"**-Prozedur aufzurufen, müssen Sie sich als **"root"** anmelden und in das **"/"** Verzeichnis wechseln.

Die Prozedur sendet eine Warnung an alle angemeldeten Benutzer.

Beispiele:

a) In 10 Minuten herunterfahren und ausschalten:

shutdown -y -g 600

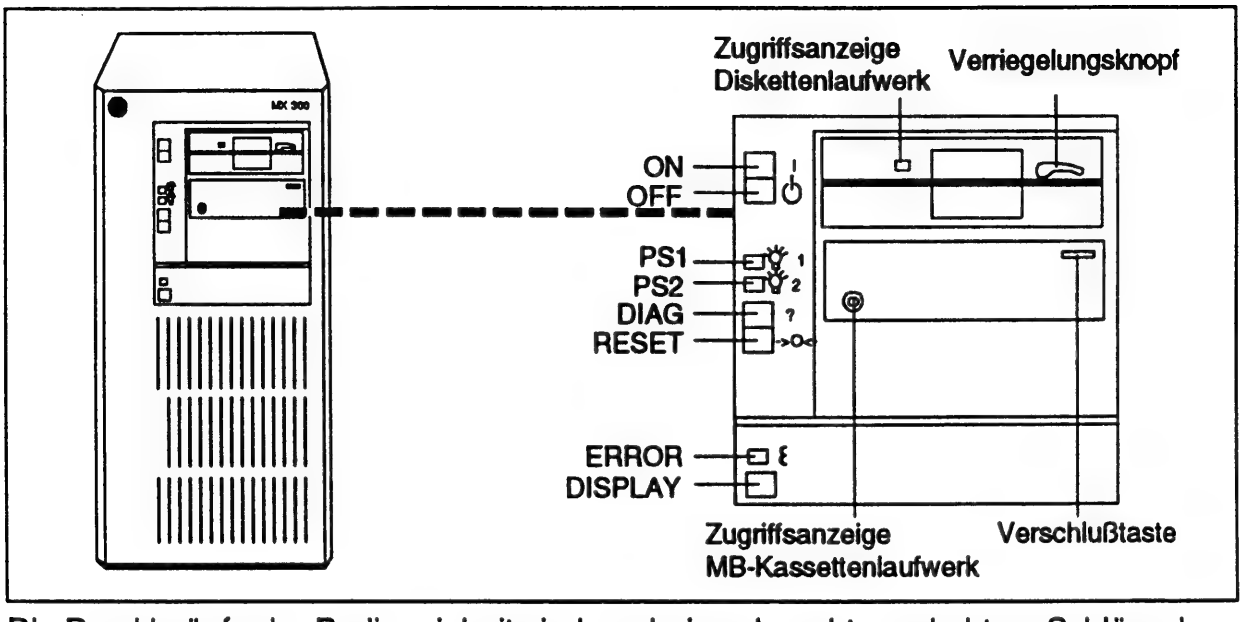
b) In 3 Minuten einen Reboot durchführen:

shutdown -y -g 180 -i6

c) Sofort in den Einbenutzerbetrieb herunterfahren:

shutdown -y -g0 u g0

Bedienfeld MX300



Die Druckknöpfe der Bedieneinheit sind nur bei nach rechts gedrehtem Schlüssel funktionsfähig. Eine offene Frontklappe bedeutet noch nicht, daß der Schlüssel in der Stellung "offen" ist.

Druckknopf "ON"
Schaltet den MX 300 ein. Vorher sollte die Konsole eingeschaltet werden, da sonst der Rechner eventuell nicht hochfährt.

Druckknopf "OFF"
Niemals betätigen, sonst können Dateien verloren gehen oder zerstört werden!

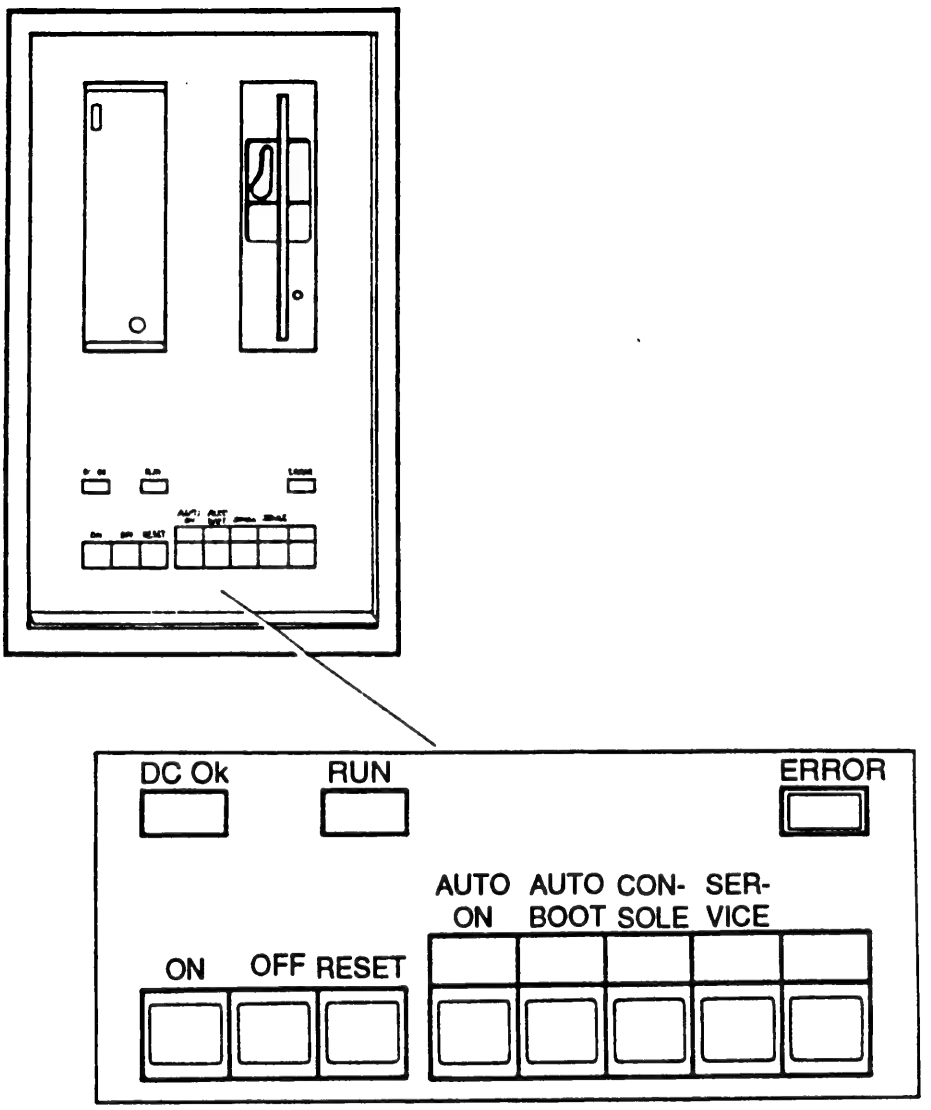
Druckknopf "DIAG"
Dient zu Diagnosezwecken. Soll ein Systemdump gezogen werden, so ist erst der Druckknopf "DIAG" und dann der Druckknopf "RESET" zu drücken.

Druckknopf "RESET"
Ist nur in Fehlerfällen (siehe Kapitel 5) zu drücken, da Daten verlorengehen können.

Leuchtdiode "ERROR"
Leuchtet, wenn z.B. während des Einschaltvorgangs ein Hardwarefehler aufgetreten ist.

Zweistelliges Display
Betriebsanzeige. Leuchtet die gelbe Leuchtdiode auf, so wird hier der aufgetretene Fehler codiert angezeigt. Während SINIX läuft, wird zyklisch gezählt. Je nach Auslastung schneller oder langsamer (Idle-Lampe). Bei hoher Auslastung bleibt die Anzeige stehen.

Bedienfeld MX500



AUTO-ON
Anlage wird nach Stromausfall automatisch wieder eingeschaltet

AUTO-BOOT
Nach dem Einschalten oder RESET wird automatisch gebootet.
Der MX500 bleibt nicht im Monitor-Modus.

CONSOLE
Konsolterminal am Steckplatz C

SERVICE
Konsolterminal am Steckplatz S

Verhalten im Fehlerfall/Notfall beim MX300:

Falls Sie sich an keinen Terminal mehr anmelden können und "nichts mehr läuft", gehen Sie wie folgt vor:

- drücken Sie die "?"-Taste (Diagnose-Monitor)
- warten Sie, bis der Rechner jegliche Aktivität eingestellt hat
- geben Sie an der Tastatur <RETURN> ein und warten Sie bis die Meldung erscheint:

  syncing disk ... done

- jetzt drücken Sie RESET oder OFF

Wenn das Betriebssystem wieder hochfährt, erzeugt es einen Arbeitsspeicherabzug unter "/var/crash", den Sie wieder löschen müssen.

rc-Dateien in V5.4

Beim Übergang in einen neuen run-level werden viele kleine produktspezifische Prozeduren in den Dateiverzeichnissen "/etc/rc0.d", "/etc/rc1.d", ... gestartet:

run-leve	Prozedur + Parameter	Dateiverzeichnis mit Prozeduren
0	/sbin/rc0 off	/etc/rc0.d
1	-> /sbin/rc1	-> /etc/rc1.d
2	/sbin/rc2	/etc/rc2.d /etc/idrc.d ==>hd
3	/sbin/rc2 /sbin/rc3	/etc/rc2.d,/etc/idrc.d /etc/rc3.d
5	/sbin/rc0 reboot	/etc/rc0.d
6	/sbin/rc6 reboot	/etc/rc0.d

Prozeduren in /etc/rc?.d:

SnnNAME — Prozeduren zum Starten von Prozessen

KnnNAME — Prozeduren zum Beenden (Kill) von Prozessen

Produkt-NAME (z.B. **"nfs"**, **"lpr"**, ...)

nn - Nummer bestimmt die Aufrufreihenfolge

Prozeduren, deren Name mit **"K"** beginnt, werden mit dem Parameter **"stop"** aufgerufen, Prozeduren, deren Name mit **"S"** beginnt, mit dem Parameter **"start"**.

Prozedur /etc/idrc.d/hd: Zugriffsrechte für Platten

Die Zugriffsrechte aller Plattengeräte aus der Datei "/etc/conf/node.d/hd" werden bei jedem Systemstart auf "600" gesetzt! Dies führt zu Fehlern bei Informix-Turbo bzw. Informix-ONLINE.

Übersicht über einige rc-Prozeduren unter /etc/rc2.d:

S01MOUNTFSYS	Dateisysteme prüfen und montieren, Zugriffsrechte von /tmp setzen
S02PRESERVE	für "vi"-Editor
S03savecore	Arbeitsspeicherabzug sichern
S05RMTMPFILES	Falls **"/tmp"** montiert ist, wird **"/tmp"** nicht **bereinigt! "/var/tmp"** wird bereinigt
S11uname	Rechnernamen (hostname) setzen
S15mkdtab	— tut nichts
S18setuname	Systemnamen setzen (SINIX-L)
S18silsd	SINIX License Daemon starten
S20klog	Protokollierung der Systemkernmeldungen in "/var/adm/klog.msg" veranlassen
...	
S69inet	LAN1 starten : /etc/inet/rc.inet Konfigurationsdatei : /etc/default/inet
S70SAFEchk	SINIX-Lizenz überprüfen (Key-Diskette?)
...	
S75cron	cron starten
S75rpc	Kommunikationsschnittstelle für LAN1
S76yp	globale Netzverwaltung für LAN1
S80lp	AT&T Spoolsystem (unnötig)
S80lpr	SINIX Spoolsystem
...	

Beispiel: rc-Prozedur "addswap", die das Kommando **"swap"** aufruft

tatsächliche Länge Ihres swap-Bereichs

 # 2. "swap"-Bereich aktivieren:

```
/usr/sbin/swap -a /dev/dsk/c1d0s2 0 66300
```

Erzeugen Sie einen Verweis dieser Prozedur unter dem Namen **"S10addswap"** im Verzeichnis /etc/rc2.d.

Beim MX500-90 existiert bereits eine solche Prozedur!
Sie tragen die swap-Bereiche nun in die Datei "/etc/swaptab" ein.

Fehler im Dateisystem

Bei jedem Systemstart werden alle Dateisysteme mit **"fsck"** überprüft:

- oberflächliche Prüfung aller Dateisysteme mit **"fsck -m"** (geht sehr schnell)
- die fehlerhaften Dateisysteme werden automatisch mit **"fsck -y"** repariert

Wie kann es zu Fehlern in den Dateisystemen kommen ?

1) Stromausfall oder unsachgemäßes Abschalten des Rechners
(nicht mit **"shutdown"** heruntergefahren)

   ⇨ kann zu gefährlichen Folgefehlern führen, wenn die Dateisysteme anschließend nicht sofort repariert werden

2) Hardwarefehler (Plattensteuerung oder Plattenfehler)
Hardwarefehler treten zunächst sehr sporadisch auf !

> Falls Sie an Ihrem Rechner Hardwarefehler vermuten, sollten Sie den Rechner täglich herunterfahren (Ausbreitung von Fehlern vermeiden).

3) **Fehler des Systemverwalters:**

   z.B. fsck auf montierte Dateisysteme im Mehrbenutzer-Betrieb oder Konfigurationsfehler

4) unerkannte Fehler des SINIX-Systemkerns
(sollte eigentlich nicht vorkommen !)

Einbenutzerbetrieb: run-level 1

Der Systemverwalter fährt den Rechner in den Einbenutzerbetrieb, weil er bestimmte Aufgaben nicht im laufenden Betrieb durchführen darf:

- Datensicherung des Betriebssystems
- Überprüfung und Reparatur von Dateisystemen (**"fsck"**)
- Dateisysteme umorganisieren (z.B. auf eine 2. Platte kopieren)
- ...

Der Systemverwalter (**"root"**) hat 2 Möglichkeiten um an der Konsole in den **"run-level 1"** herunterzufahren:

1) `shutdown -y -g60 -i1`

2)
```
wall ...      # Benutzer warnen
init 1        # sofort und ohne Warnung herunterfahren
sleep 60      # ca. 60 Sekunden warten
killall 15    # allen Hintergrundprozessen Signal 15
              # senden
sleep 10      # ca. 10 Sekunden warten
killall       # allen Hintergrundprozessen Signal 9
              # senden
umountall     # Dateisysteme bis auf /, /stand und /proc abmontieren
```

Warum müssen Sie Hintergrundprozesse beenden, um Dateisysteme mit **"umountall"** abzumontieren ?

"fsck" selbst aufrufen ?

- **rufen Sie "fsck" auf, wenn Sie den Verdacht haben, daß Ihre Dateisysteme nicht in Ordnung sind**
- **"fsck" nur auf abmontierte Dateisysteme anwenden (vor "fsck" unbedingt "umount" bzw. "umountall") !!!**
- **"root"-Bereich nur im "run-level 1" überprüfen !**

- **als "root" an der Konsole anmelden**
- `shutdown -y -g120 -i1`
- warten bis System im Einbenutzerbetrieb (3 Minuten)
-
```
killall 15      # Prozesse möglichst geordnet beenden
sleep 10
killall
sleep 2
```
-
```
umountall       # alles abmontieren bis auf /stand, /proc
mount           # prüfen, ob wirklich alles abmontiert ist
```
- "fsck" nacheinander auf alle Dateisysteme, die in der /etc/vfstab stehen:

```
sync ; sync
fsck -y /dev/rdsk/c0d0s3     # "usr"-Dateisystem
fsck -y /dev/rdsk/c0d0s4     # "home"
fsck -y /dev/rdsk/c0d0s11    # "var"
fsck -y /dev/rdsk/c0d0s12    # "opt"
fsck -y /dev/rdsk/c0d0s13    # "tmp"
      ...
```

- Überprüfung des "root"-Dateisystems:

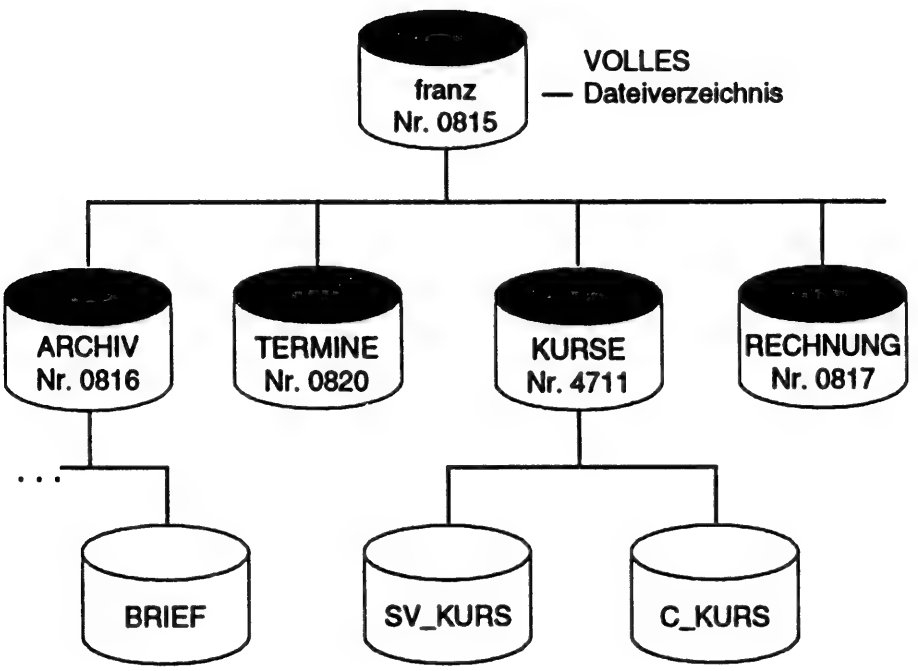
```
fsf off
fsck -y /dev/rdsk/c0d0s1          # "root" Dateisystem
```

Falls Fehler aufgetreten sind, drücken Sie nun: **RESET**-Taste (ohne sync !!)
Andernfalls: `init 6` # neu hochfahren

lost+found

"fsck" muß eventuell Dateien oder Dateiverzeichnisse **"wegreparieren"**, um die Konsistenz des Dateisystems wieder herzustellen !

Was passiert, wenn **"fsck"** ein volles Dateiverzeichnis **"wegrepariert"** ?



Regelmäßige Kontrollaufgaben

Protokolldateien kontrollieren und zurücksetzen:

In jedem UNIX-System werden aus Sicherheitsgründen zahlreiche Protokolldateien angelegt, die der Systemverwalter regelmäßig kontrollieren und zurücksetzen sollte. Andernfalls wachsen diese Protokolldateien immer mehr an und belegen viel Plattenplatz.

Sie können eine Protokolldatei auf 0 Bytes zurücksetzen, indem Sie einfach die Ausgabeumlenkung der Shell verwenden:

Beispiel: **> /var/adm/klog.msg**

Protokolldateien stehen in den Dateiverzeichnissen **"/var/adm"** und **"/var/cron"**.

Es existiert eine Prozedur **"/sbin/cleanup"**, die einige Protokolldateien automatisch zurücksetzt. Diese Prozedur sollten Sie erweitern (siehe Anhang).

Sie müssen folgende Dateien regelmäßig zurücksetzen:

/var/adm/klog.msg	Meldungen des Systemkerns
/var/adm/lastlog	Datum der letzten Anmeldung für jeden Benutzer
/var/adm/sulog	Protokoll der "su"-Aufrufe zum Superuser
/var/adm/wtmp	Protokoll aller Anmeldungen lesen mit: who /var/adm/wtmp
/var/adm/wtmpx	**erweitertes Protokoll aller Anmeldungen (wird sehr groß!!)**
/var/cron/log	Protokolldatei des "cron"
/var/crash/*	Arbeitsspeicherabzüge nach Systemabsturz oder Stromausfall (riesig groß !!!!)

In einigen Versionen von SINIX V5.4 werden die temporären Dateien unter "/tmp" nicht automatisch gelöscht !

Sie müssen eventuell auch unter "/tmp" aufräumen und alle Dateien löschen, die älter als 3 Tage sind.

Mehrfach-erfolglose Logins protokollieren:

/var/adm/loginlog

Allgemeines:
In SINIX V5.40 existiert ein Mechanismus, der nicht erfolgreiche "logins" mitprotokolliert. Sollte ein Anwender fünf mal erfolglos versuchen, sich an das System anzumelden und es existiert eine Datei "/var/adm/loginlog", so werden alle Fehlversuche in diese Datei protokolliert.

Die Datei "/var/adm/loginlog" ist standardmäßig nicht vorhanden, d.h. der Mechanismus ist inaktiv. Durch Anlegen der Datei nach folgendem Schema wird der Protokollmechanismus aktiviert:

"> /var/adm/loginlog"
"chgrp sys /var/adm/loginlog"
"chown root /var/adm/loginlog"
"chmod 600 /var/adm/loginlog"

Um zu vermeiden, daß die Datei mit der Zeit eine störenden Größe erreicht, ist es empfehlenswert diese gelegentlich zu löschen.

Log-Datei für X-Terminals:

/var/adm/bootplog Protokolliert jeden "boot" eines X-Terminals über bootp

Prozeduren automatisch starten

Kommandos, die regelmäßig zu bestimmten Zeiten gestartet werden sollen, können Sie in eine crontab-Datei eintragen (z.B. Datensicherung um 22.00 Uhr).
Dies ist besonders nützlich bei Rechnern, die auch nachts laufen.

crontab Kommando:

In SINIX V5.4 kann jeder berechtigte Benutzer mit dem "crontab"-Kommando eine eigene "crontab"-Datei anlegen:

```
vi mycrontab          # eigene crontab erstellen

crontab mycrontab     # eigene crontab "absenden"

crontab -l            # aktuelle crontab-Datei ansehen

crontab -r            # aktuelle crontab-Datei deaktivieren
```

crontab Dateien:

Prozeduren oder Kommandos zur Systemverwaltung sollten in der "root-crontab" stehen. Sie finden die "crontab"-Dateien der einzelnen Benutzer im Verzeichnis:

/var/spool/cron/crontabs/

Beispiel für eine "root"-crontab-Datei:

```
#ident "@(#)adm:root 1.5.1.2"
#ident "$Header: root 2.1 90/05/18 $"
10      15      * * 5    /bin/su root -c   "/sbin/cleanup > /dev/null"
48      11,14   * * 1-5  /bin/su uucp -c  "/usr/lib/uucp/uudemon.admin > /dev/null 2>&1"
40      *       * * *    /bin/su uucp -c  "/usr/lib/uucp/uudemon.poll > /dev/null"
26,56   *       * * *    /bin/su uucp -c  "/usr/lib/uucp/uudemon.hour > /dev/null"
0       2       * * 0,4  /etc/cron.d/logchecker
```

Minute(n) | Stunde(n) | Tag(e) im Monat | Monat im Jahr (0-11) | Tag in der Woche

Die Prozedur "/sbin/cleanup" soll Aufräumarbeiten unter "/var/adm" durchführen! Sie müssen allerdings diese Prozedur an Ihrem System erweitern und zu einer Zeit starten, zu der Ihr System läuft!

Das "at"-Kommando:

- startet Kommandos oder Prozeduren zu einem späteren Zeitpunkt
- **"at"** liest die Kommandos oder die Prozedur von der Standardeingabe

Beispiele:

```
at 18:10 < prozedur      # startet prozedur nach 18:10 Uhr
at -l                    # listet alle offenen Aufträge auf
```

Achtung:

- crontab-Aufträge werden nur erledigt, wenn der Rechner zum angegebenen Zeitpunkt läuft
- **"at"**-Aufträge werden auch noch nachträglich ausgeführt

Berechtigung für crontab und at

Die Berechtigung zur Verwendung des **"crontab"**-Kommandos muß der Systemverwalter über eine der beiden folgenden Dateien festlegen:

/etc/cron.d/cron.deny oder alternativ:
/etc/cron.d/cron.allow

- Ist **"cron.allow"** vorhanden, so dürfen nur die Benutzer mit **"crontab"** arbeiten, deren Name in der Datei **"cron.allow"** steht.
- Ist **"cron.allow"** nicht vorhanden sondern **"cron.deny"**, so dürfen alle Benutzer mit **"crontab"** arbeiten, deren Name nicht in der Datei **"cron.deny"** steht.

Berechtigung für "at"

Der Systemverwalter vergibt die Berechtigung für das **"at"**-Kommando über die Dateien:

/etc/cron.d/at.deny oder alternativ:
/etc/cron.d/at.allow

Prozeduren aus einer crontab starten :

- Prozeduren, die aus einer crontab gestartet werden, sollten ihre Umgebungsvariable PATH selbst setzen und exportieren:

  PATH=/sbin:/usr/bin:/usr/sbin:/opt/bin; export PATH

- Außerdem sollten Sie die Standard-Ausgabe- und Fehler-Känale auf eine Datei umlenken oder über Pipe als "mail" versenden. Dies gilt auch für "at"-Aufträge !

Aufgabe:

Sehen Sie sich die Datei /etc/inittab und die Verzeichnisse /etc/rc?.d an. Was passiert alles beim Systemstart?
Untersuchen Sie die Prozeduren bis einschließlich "S20klog".

Datensicherung

Geräte zur Datensicherung

1/4 Zoll Magnetbandkassettenlaufwerke (Streamer):

- **altes 60 MB Laufwerk mit 9 Spuren: TDC 3319 / TDC 3610**

- **155 MB Laufwerke mit 18 Spuren: TDC 3650 oder TDC 3660**

  Die 155 MB Laufwerke sind seit 09.89 wahlweise in alle neuen MX300, MX500 und WX200 Modelle eingebaut.
  Die Laufwerke arbeiten mit einer größeren Schreibdichte, daher können Sie mit diesen Laufwerken nur noch 60/155 MB Kassetten U7-H3 beschreiben.

Die 155 MB-Laufwerke können Bänder lesen, die mit 60 MB-Laufwerken beschrieben wurden.

Beschreiben Sie jedoch ein Band mit einem 155MB Laufwerk, so können Sie dieses Band auf keinem Rechner einlesen, der noch ein 60 MB-Laufwerk besitzt !!!

Magnetbandkassettenlaufwerk mit 2.0 GB (Exabyte):

Externes Gerät, das mit Video 8 (mm) Kassetten und rotierenden Schreib-/Lese-Köpfen arbeitet (Exabyte).

Für dieses Gerät benötigen Sie einen SCSI-Controller (Hostadapter) vom Typ 1, der in Ihren Rechner eingebaut sein muß. An diesen SCSI-Controller können Sie jedoch keine SCSI-Platten anschließen !

Sicherung auf externe SCSI-Platten:

Für externe SCSI-Platten (760 MB) benötigen Sie einen SCSI-Controller vom Typ 2 und einen Zusatzschrank.

Arbeiten mit Magnetbandkassette

Gerätedateien für MB-Kassetten:

a) Automatisches Zurückspulen beim "**close**" auf das Gerät:

Streamer: /dev/tape oder /dev/rmt/c0s0

Exabyte: /dev/exa0

b) Das Band wird nicht automatisch zurückgespult:

Streamer: /dev/tapen (norewind) oder /dev/rmt/c0s0n

Exabyte: /dev/exa8

Wird das Streamer-Laufwerk geöffnet, so spult der Treiber das Band vor dem nächsten Zugriff automatisch zurück.

Standard-Streamer zurückspulen und straffen:

Wird das Streamer-Laufwerk geöffnet, so spult der Treiber das Band **vor dem nächsten Zugriff** automatisch zurück und strafft das Band.
Straffen bedeutet, das Band bis zum Ende vor- und zurückzuspulen (Dauer: ca. 2 Minuten). Dieser Vorgang ist wichtig für die Sicherheit des späteren Lese- oder Schreibvorgangs (siehe "**mt**": Bandkontrolle).

Falls Sie beim Einlesen eines Bandes Lesefehler erhalten, legen Sie das Band erneut ein, lassen das Band erneut straffen und zurückspulen und versuchen es noch einmal.

Pufferung

In SINIX V5.x puffert der Systemkern jede Streamer-Ein-/Ausgabe automatisch mit einem 1 MB-Puffer und vermeidet damit einen langsamen Start-/Stop-Betrieb.

Blockungsfaktor:

Viele Sicherungskommandos besitzen Optionen, mit denen Sie die Größe der Datenblöcke angeben können, die das Kommando bei Ein- bzw. Ausgabe verwendet:

Beispiele:

tar :	Option	-b		
dd :	Option	bs=		
cpio:	Optionen	-C	bzw.	-B

Die Ein-/Ausgaben auf den Streamer werden in SINIX V5.x vom Systemkern gepuffert, so daß Sie normalerweise keinen "Blockungsfaktor" angeben müssen.

Bandendeerkennung:

Die automatische Pufferung des Systemkerns hat einen Nachteil:

Kein Kommando ist in der Lage, automatisch das Bandende festzustellen (auch nicht "cpio") !

Warum nicht ?

Einige Kommandos besitzen Optionen, mit denen Sie dem Kommando bereits bei Aufruf die Bandkapazität mitteilen können.

Beim Exabyte können Sie zwischen gepufferter und ungegepufferter Ein-/Ausgabe wählen.

Bei der **ungepufferten Ein-/Ausgabe** müssen Sie einen "Blockungsfaktor" angeben: cpio -C...
Die maximale Blockgröße des "privaten" Puffers ist bei SINIX V5.4 allerdings auf 32 KB begrenzt.

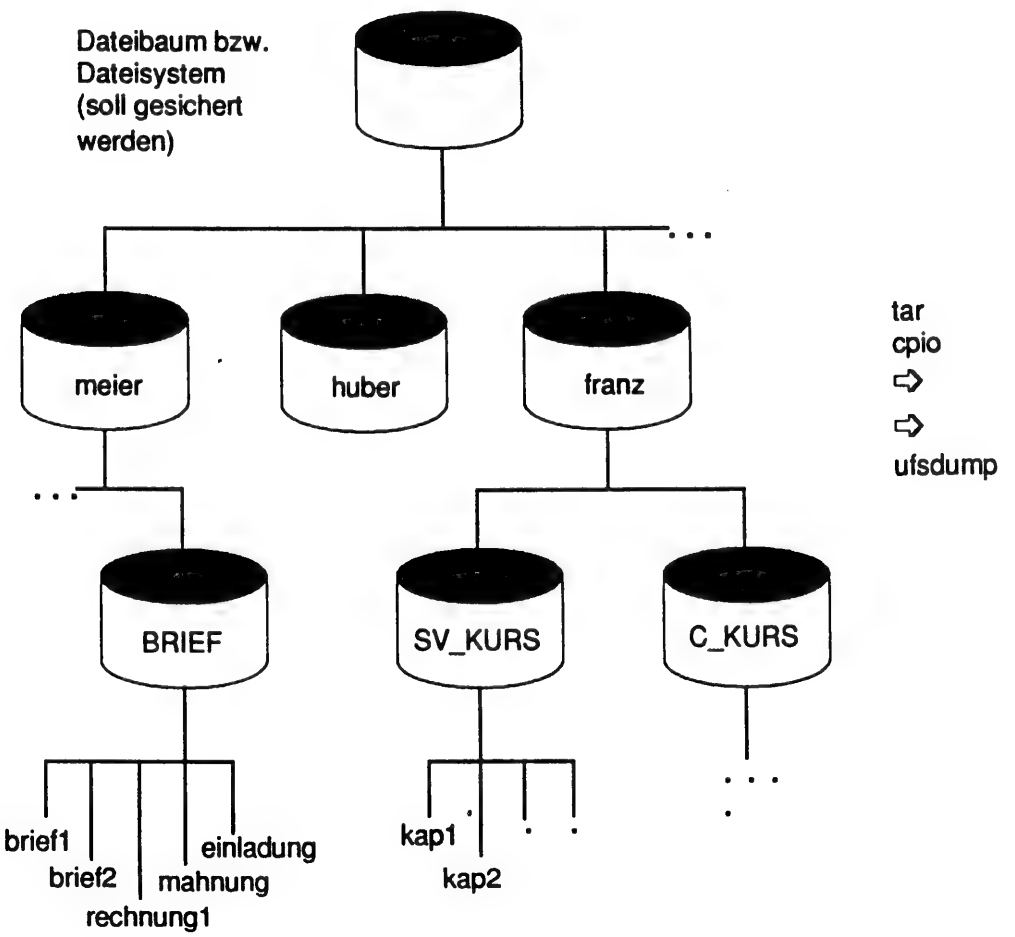
Grundprinzip von tar, cpio und ufsdump

Die Programme **"tar"**, **"cpio"** und **"ufsdump"** dienen unter anderem zur Sicherung von Daten auf einem Magnetband.

tar, cpio **sichern beliebige Dateibäume**
oder einzelne Dateien

ufsdump **sichert komplette "ufs"-Dateisysteme (nicht "s5" oder"bfs")**
⇨ Einlesen mit: **ufsrestore**

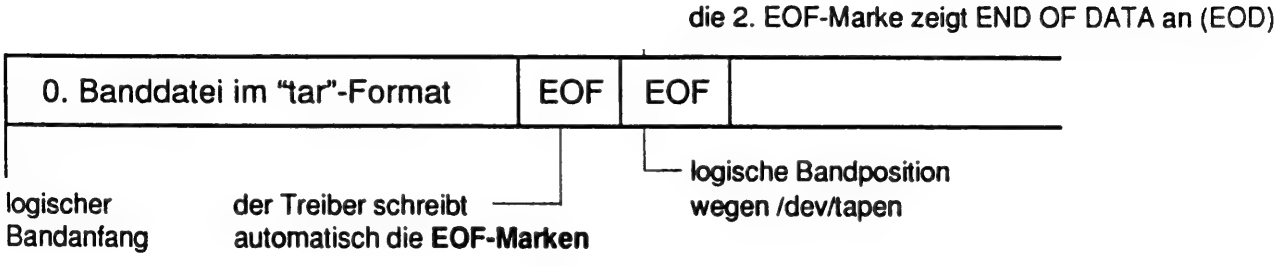
Was machen diese Programme mit den Daten ?



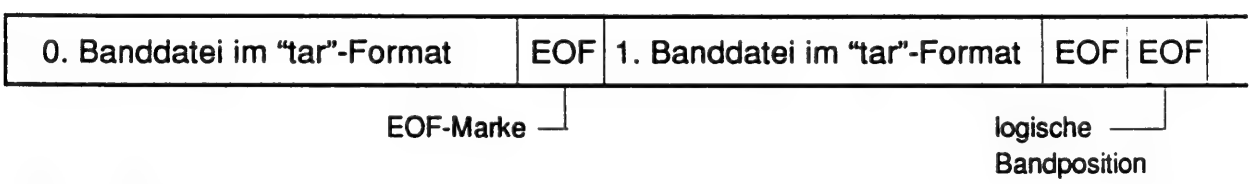
Erzeugen mehrerer Banddateien

Beispiel:

```
cd /home/meier
tar cvf /dev/tapen .      # tar erzeugt eine große Datei
                          # und schreibt sie auf das Band
```




```
cd /home1/mueller
tar cvf /dev/tapen .
```




```
echo hallo > /dev/tape
```

Was passiert mit den Daten, wenn Sie jetzt eingeben :

```
echo Unsinn >/dev/tapen ???
```

"mt": Bandkontrolle

Sie sollten bei der Installation des Betriebssystems auch das Paket "compat" (compatibility package) installieren. Sie können das Paket auch nachträglich installieren.

Wenn "compat" installiert ist, haben Sie unter "/usr/ucb" viele nützliche "Berkeley"-Kommandos zur Verfügung, unter anderem das Kommando "mt":

```
/usr/ucb/mt -f /dev/tape noret      # Straffen vermeiden

/usr/ucb/mt -f /dev/tape ret        # Magnetband straffen
```

Löschen/ Positionieren

```
/usr/ucb/mt -f /dev/tape erase      # das ganze Band löschen
                                    # (mit Löschkopf: 2 Min, bei
                                      Exabyte: Stunden)
echo Unsinn > /dev/tape             # ab der aktuellen
                                    # Stelle löschen

/usr/ucb/mt -f /dev/tapen fsf       # um eine Banddatei
                                    # vorwärts spulen

/usr/ucb/mt -f /dev/tapen fsf 2     # um zwei Banddateien
                                    # vorwärts spulen

< /dev/tape                         # an Bandanfang zurückspulen
```

Zurückpositionieren:

Mit dem Standard-Streamer ist es nicht möglich, um eine Banddatei zurückzupositionieren !

Wollen Sie mit dem 2.0 GB Magnetband-Kassetten-Laufwerk (Exabyte) um **eine Banddatei zurückspulen,** müssen Sie wie folgt vorgehen:

2 EOF-Marken zurück : `/usr/ucb/mt -f /dev/exa8 bsf 2`
1 EOF-Marke vorwärts: `/usr/ucb/mt -f /dev/exa8 fsf 1`

Sie können ähnliche Funktionen auch mit dem Kommando **"tapecntl"** ausführen.

Physikalische Einteilung des Standard-Streamerbandes

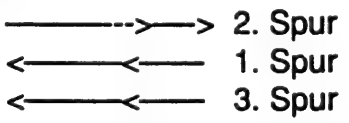
Physikalisch beschreiben die Standard-Streamer-Laufwerke nacheinander mehrere Spuren auf dem Magnetband. Jede Spur wird seriell beschrieben bzw. gelesen.

Das Magnetband wird dabei abwechselnd in beiden Richtungen transportiert.

> Die physikalische Bandposition stimmt daher nicht mit der logischen Bandposition überein.

Beispielschema :

Streamerband



Bänder kopieren ?

Da eine Magnetbandkassette sehr viele Banddateien enthalten kann, ist es nicht einfach, ein Band zu kopieren. Mit dem Kommandos "cp" oder "dd" können Sie jeweils nur eine Banddatei kopieren !

Beispiel:

```
cp /dev/tapen band.0        # kopiert die 0. Banddatei auf
                            # die Platte
cp /dev/tapen band.1        # kopiert die 1. Banddatei auf
                            # die Platte
    ...
```

Wenn Sie genügend Platz auf Ihrer Platte haben, können Sie zum Kopieren von Bändern folgende Prozeduren verwenden:

Beispiel:

Prozedur "cpin": von Band auf Platte

```
#!/bin/sh
# Allgemeine Prozedur zum Kopieren von
# Streamer-Band auf Platte: cpin
# (mit "cpout" wieder herausschreiben)
#
# Band straffen und zurueckspulen
/usr/ucb/mt -f /dev/tape ret
#
# 0. Banddatei kopieren
cp /dev/tapen band.0

#
# Alle anderen Banddateien kopieren
i=0
while test -s band.$i
do
    ls -l band.$i
    i=`expr $i + 1`
    if cp /dev/tapen band.$i
    then :
    else
        echo "Fehler oder Bandende !!"
# am Bandende kommen ev. Fehlermeldungen!
        break
    fi
done

# Band zurueckspulen
/usr/ucb/mt -f /dev/tape rewind
```

Prozedur "cpout": von Platte auf Band

```
#!/bin/sh
# Allgemeine Prozedur zum Kopieren von
# Platte auf Band: cpout
# (Dateien vorher mit "cpin" einlesen !)
#
# Band straffen und zurueckspulen
 /usr/ucb/mt -f /dev/tape ret
#



#
# Alle  Banddateien kopieren
i=0
while test -s band.$i
do
    ls -l band.$i
    if cp band.$i /dev/tapen
    then :
    else
        echo "Fehler beim Kopieren !!"
        exit 1 # bei Fehler abbrechen
    fi
    i=`expr $i + 1`
done
#
#
# Band zurueckspulen
/usr/ucb/mt -f /dev/tape rewind
```

Sicherungskommandos

Kommando	Was kann gesichert werden ?	Geräte/ symbol. Links ?	Differenz-Sicherung?	Berücksichtigung der Bandlänge ?
tar	Dateibäume von "s5", "ufs" und "bfs"	nur in SINIX V5.4 !	mit "find ... -newer .." möglich	mit Schalter -k nnn (nnn in KB)
cpio mit find	Dateibäume von "s5", "ufs" und "bfs"	ja mit Schalter -c	mit "find ... -newer .." möglich !	Bandendeerken-kennung funkt-tioniert nicht wegen gepufferter Ausgabe
dd und cp	physikalische Sicherung von beliebigen "slices"	ja	nein	nein
ufsdump ufsrestore	nur "ufs"-Dateisysteme	ja	eingebaute Funktion	ja (Folgebänder)

- UNIX System V.4 bietet zusätzlich ein "**backup**"-System an, das sehr komplex ist und wahlweise über "**sysadm**" oder folgende Kommandos bedient werden kann:

  - backup (1M)
  - bkhistory (1M)
  - bkoper (1M)
  - bkreg (1M)

  ...

  funktioniert nicht
  Meldung: service not available

- In SINIX V5.4 bietet Ihnen das Produkt "**DSX**" ein menügeführtes Datensicherungssystem.

  Der Nachteil komplexer, menügeführter Sicherungssysteme besteht jedoch darin, daß im "**Ernstfall**" (Datenverluste, defekte Platte, defekte Dateisysteme) zunächst das Sicherungssystem restauriert werden muß.

  Wir wollen auch die Möglichkeiten zur Datensicherung betrachten, mit denen wir das System nach einem "**Ernstfall**" notfalls mit den "**boot**"-Disketten oder mit dem Systemband wieder restaurieren können.

cpio

1) **Dateibaum kopieren: cpio -p**

Beispiel: /home/meier —> /home1/mueller

```
cd /home/meier      # Quelle        Ziel:
find . -print       | cpio -pdm /home1/mueller
```

2) **Dateibäume sichern: cpio -o**

```
cd /home/meier      # Quelle

find . -print       | cpio -oVBc > /dev/tape
```

Wollen Sie Dateien mit "cpio" kopieren oder sichern, müssen Sie die Dateinamen über die Standardeingabe an "cpio" übergeben.

3) **cpio-Sicherungen einlesen: cpio -i**

```
cd /home1/mueller

cpio -ivBdmu ['MUSTER'] < /dev/tape
```

Inhaltsverzeichnis: `cpio -itv < /dev/tape`

Bedeutung der verwendeten Schalter:

- B Blockgröße 5120 Bytes (Standard sind 512 Bytes)
- v Dateinamen und Meldungen ausgeben
- V für jede Datei einen Punkt ausgeben
- d notwendige Dateiverzeichnisse anlegen
- m das Änderungsdatum bleibt erhalten
- u neuere Dateien beim Einlesen überschreiben
- c "portables" Format (ab V5.4: Prüfsummen, Pfadnamen bis 1024 Bytes, nicht kompatibel zu älteren UNIX-Versionen)
- **H odc** entspricht dem "-c"-Format älterer UNIX Versionen (z.B.: SINIX V5.22)

Dateibaum kopieren

Teile des Dateibaums so kopieren, daß die Zugriffsrechte und die Modifikationszeiten der Dateien gleich bleiben.

Beispiel:

Sie wollen den Benutzer **"meier"**, der bisher im **"/home"**-Dateisystem arbeitet, in das Dateisystem "/**home1**" umsiedeln. Der Benutzer **"meier"** soll von dieser Aktion möglichst nichts merken (keine Nachteile haben).

⇨ Achtung

In SINIX V5.2x ist **"cpio"** nicht geeignet um den **"root"** oder den **"usr"**-Bereich zu sichern oder zu kopieren !

In SINIX V5.4 können Sie **"cpio"** für alle Zwecke verwenden!

Gerätedateien für Disketten in V5.4

1. Laufwerk: 3,5 Zoll (Standard)

- high density : 1,4 MB Kapazität

AT&T-Gerätedatei : /dev/rdsk/**f03ht** | ohne Zylinder 0: f03h

XENIX-Gerätedatei : /dev/**rfd0135ds18** (135 tpi, 18 Sekt./Spur)

- double density : 720 KB Kapazität

AT&T-Gerätedatei : /dev/rdsk/**f03dt** | ohne Zylinder 0: f03d

2. Laufwerk: 5 1/4 Zoll (optional)

- high density : 1,2 MB Kapazität

AT&T-Gerätedatei : /dev/rdsk/**f15ht** | ohne Zylinder 0: f15h

XENIX-Gerätedatei : /dev/**rfd196ds15** (96 tpi, 15 Sektoren/Spur)

fl2-Bereich von SINIX V5.2x: /dev/**rdsk/tffl2h** (nur lesen)

- quad density : 720 KB Kapazität (double density mit 96 tpi)

AT&T-Gerätedatei : /dev/rdsk/**f15qt** | ohne Zylinder 0: f15q

fl2-Bereich von SINIX V5.2x: /dev/rdsk/**tffl2** (nur lesen)

Gerätedateien, die das Format automatisch erkennen:	
1. Laufwerk : /dev/rdsk/**f0t**	ohne Zylinder 0: /dev/rdsk/f0
2. Laufwerk : /dev/rdsk/**f1t**	ohne Zylinder 0: /dev/rdsk/f1

Datenaustausch mit SINIX V5.2x

SINIX V5.4 unterstützt nicht mehr das ehemalige SINIX-Disketten-Format mit den Bereichen:

	fl1	/dev/rfl2 (in SINIX V5.2x)	

nicht bei SINIX V5.4 !!!

fl0

Falls Sie das optionale 5 1/4 Zoll Laufwerk besitzen, können Sie den "fl2"-Bereich von V5.2x-Disketten lesen aber nicht beschreiben:

/dev/rdsk/tffl2 bei Standard-Disketten
/dev/rdsk/tffl2h bei high density Disketten

Der Datenaustausch über das optionale 5 1/4 Zoll Laufwerk funktioniert am besten, wenn Sie das 720 KB Format und "double density" Disketten mit 96 tpi verwenden:

/dev/rfl3 720 KB (in SINIX V5.2x)

/dev/rdsk/f15qt 720 KB (in SINIX V5.4)

5 1/4 Zoll:	SINIX/UNIX V5.4	SINIX V5.2x
Formatieren:	format /dev/rdsk/f15qt	/etc/flformat -s 9 -b 512 -f N
Schreiben:	tar cvf /dev/rdsk/f15qt ...	tar cvf /dev/rfl3 ...
Lesen:	tar xvf /dev/rdsk/f15qt	att tar xvf /dev/rfl3 # Problem bei SINIX V5.21

Sie können statt "tar" auch das "cpio"-Kommando zum Datenaustausch verwenden (jedoch nicht mit Schalter "-c" !)

Wenn Ihr SINIX V5.2x Rechner ein 150 MB Streamerlaufwerk besitzt, tauschen Sie Ihre Daten am besten über Magnetbandkassetten aus.

Das tar Kommando

"**tar**" archiviert Dateien oder Dateibäume in einer "**tar**"-Datei. Die Datei kann auf Diskette, auf einem Magnetband oder auf der Platte liegen.

Unterschiedliche tar-Versionen

SINIX V5.2x		SINIX V5.4
"ucb"-tar	ab SINIX V5.22: neuer "att"-tar	ganz neuer "tar"
X/Open P. Guide 2 (XPG 2)	X/Open P. Guide 3 (XPG 3)	X/Open P. Guide 3 (XPG 3)
+ SINIX-Erweiterungen	ohne Erweiterungen	+ SINIX-Erweiterungen
		+ XENIX-Erweiterungen
sichert Pfadnamen bis 100 Bytes	sichert Pfadnamen bis 255 Bytes	sichert Pfadnamen bis 255 Bytes
keine Gerätedateien	sichert Gerätedateien	sichert Gerätedateien
steht auch im "sie"-Universum zur Verfügung		
"sie far" bzw. "ftar" für Disketten	ruft bei XPG2-Format automatisch den "ucb"-tar auf	kann XPG2-Format lesen

Die verschiedenen "**tar**"-Kommandos unterscheiden sich in vielen Punkten, u.a. in der Bedeutung der Schalter "**p**", "**o**" und "**k**".

Datensicherung

tar Aufruf:

Daten sichern (auf Band / Diskette schreiben) :

```
cd /home
tar cvf /dev/tape
```

Liste von Dateien / DVZ

```
mueller meier franz
```

schreiben — Streamer

auf Diskette:

```
tar cvfb /dev/rdsk/f0t  20  mueller meier franz
```

"Blockungsfaktor", Pufferung vom tar
⇨ Geschwindigkeit

hinten anhängen (nur bei Disketten):

```
tar rvnfb /dev/rdsk/f0t  20 huber
```

Falls Sie nicht sicher sind, ob Ihre Daten auf das Band / die Diskette passen:

```
tar cvkf 150000 /dev/tape .
```

Bandlänge in KB — aktuelles DVZ

Kombination aller Schalter:

```
tar cvkfb 1440 /dev/rdsk/f0t  20  meier
```

Schreibarbeit sparen, durch Datei:

/etc/default/tar

Sie müssen in diese Datei sinnvolle Werte eintragen:

Beispiel:

	block	Länge
archive0=/dev/rdsk/f0t	20	1440
archive1=/dev/rdsk/f1t	20	1200
archive2=/dev/rdsk/tffl2	20	584
archive3=/dev/rdsk/f15qt	20	720
archive4=/dev/tape	20	150000

tar Aufruf mit Bezug auf /etc/default/tar

```
tar cv0   meier mueller franz

tar xv2   # einlesen von tffl2!

tar rvn0 huber
```

anhängen!

⇨ **Achtung:**

Falls eine Umgebungsvariable TAPE gesetzt ist, verwendet "tar" den Inhalt der Variablen als Gerät. In diesem Fall liest "tar" nicht die /etc/default/tar.

tar

Vor- und Nachteile von "tar"

+ **einfache Bedienung:** Sie müssen keine Liste von Dateinamen über **"find"** erzeugen

Beispiel: Sie wollen **"/home/meier"** und **"/home/mueller"** sichern.
Wie lauten die Kommandos für **"tar"**, wie für **"cpio"**?

+ **Berücksichtigung der Bandlänge** über den Schalter "-k" (eventuelle Folgebandverarbeitung)

- "boot"-Disketten enthalten nicht "tar" sondern "cpio"

- Die Liste der Dateinamen, die Sie als Argument übergeben können, ist systembedingt auf ca. 5 KB begrenzt.
  Dies führt zu Problemen bei Differenzsicherungen.
  Warum ?

Deshalb: Spezielle Option "-I" bei SINIX V5.4:

```
tar cvf /dev/tape -I list_dat
```

Mit dieser Option sind auch umfangreiche Differenzsicherungen möglich.

dd

Das Kommando **"dd"** ist ein Kopierprogramm (ähnlich "cp"), mit dem Sie einzelne Dateien oder komplette "slices" aus UNIX-Sicht physikalisch kopieren können.

Sie können z.B. "dd" benutzen, um:

a) die Dateisysteme "/", "/stand" und "/usr" in gleich große "slices" einer anderen Platte zu kopieren, damit im Notfall ein Ersatzsystem zur Verfügung steht

b) "slices" physikalisch auf Magnetbandkassette zu schreiben

c) Dateiinhalte oder Teile von Dateien ausgeben zu lassen

Einige Optionen von "dd":

if=ein_dat	Eingabedatei (Standard: Standardeingabe)
of=aus_dat	Ausgabedatei (Standard: Standardausgabe)
bs=n	Blockgröße für Eingabe- und Ausgabeblöcke
skip=n	die ersten n Eingabeblöcke überspringen
seek=n	in der Ausgabedatei n Blöcke überspringen
count=n	nur n Blöcke kopieren (Std: bis Dateiende)

"dd"-Aufrufe für die obigen Beispiele:

a) Vorbereitung: ???

Kopieren des "/"-Bereichs:

b) wie a) aber: of=/dev/tape

c)
```
dd if=/etc/passwd bs=128 count=1
```

Sicherungskonzept

Sicherung der Benutzerdaten:

Dateisysteme: /home, /home1, ...

Die Benutzerdaten können Sie mit beliebigen Kommandos (z.B. "tar" oder "ufsdump") sichern. Für Benutzerdaten bieten sich auch menügeführte Systeme an, wie z.B. "DSX".

Sicherung des Betriebssystems:

Dateisysteme: /stand, /, /usr, /var, /opt

Das Betriebssystem muß so gesichert werden, daß Sie es im Notfall (z.B. Ausfall einer Platte) von den Boot-Disketten oder vom Minirootsystem einlesen können. Sie müssen sich daher an folgende Regeln halten:

- Sicherungen mit "cpio -oBc" und zwar mit **relativen** Pfadnamen !

- Wichtig: Vor der Sicherung in den Einbenutzer-Betrieb wechseln und /dev/fd und /proc abmontieren !

- Die Platteneinteilungen (Slice-Größen) müssen separat gesichert werden, damit Sie eine Ersatzplatte neu einteilen können.

- Bereinigen Sie vor der Sicherung die Dateiverzeichnisse "/var/crash", "/var/adm", "/var/cron", ...

- Sie müssen vorher prüfen, ob das gesamte System auf ein Band paßt. Andernfalls müssen Sie z.B. "/opt" auf einem 2. Band sichern.

  Typischer Platzbedarf der Dateisysteme:

stand	1,3 MB	
root	9,0 MB	
usr	50 - 60 MB	
var	5,0 MB	(vorher bereinigen !!)
opt	50 - 60 MB	
Summe:	**116- 136 MB**	

Einem 155 MB-Band können Sie ca. 135-140 MB Nettodaten anvertrauen. Der Platzbedarf läßt sich nicht genau bestimmen, da Dateien mit mehreren "Hardlinks" vom "cpio" mehrfach gesichert werden.

Datensicherung

Arbeitsschritte zur Systemsicherung:

- in run level 1 herunterfahren

- alle Hintergrundprozesse beenden

- "/var/crash", "/var/adm" und "/var/cron" bereinigen (z.B. mit eigener Prozedur "cleanup")

- Platteneinteilungen mit "prtvtoc -f vtoc1 /dev/rdsk/..." lesen und mit "/etc/vfstab" vergleichen.
  Am besten erzeugen Sie für jede Platte eine Datei, die als Eingabedatei für "disksetup -Id ..." dienen kann. In Anhang des Kursordners finden Sie eine Prozedur "save_vtoc", die solche Dateien automatisch erzeugt.

  Die Ausgabedateien von "prtvtoc -f ..." und die Dateien, die als Eingabedatei für "disksetup" geeignet sind, sichern Sie mit "cpio" in die **0. Banddatei.**

- cd /
  Dateisysteme "/dev/fd" und "/proc" abmontieren

- "/" und "/stand" mit "cpio" und relativen Pfadnamen in **1. Banddatei** sichern

- "/usr" und "/var" mit "cpio" und relativen Pfadnamen in **2. Banddatei** sichern

- Falls "/opt" nicht mehr als 60 MB belegt, können Sie "/opt" in die **3. Banddatei** sichern

- Sie können die Systemdateien unter "/home" noch in die **4. Banddatei** sichern:

  /home/oasys, /home/vmsys, /home/tele, /home/dsx

- den Rechner mit "init 6" neu hochfahren

  Die Sicherung der Benutzerdaten sollten Sie auf separaten Bändern durchführen.

Grundsystem sichern

Aufgabe: 5

a) Entwerfen Sie eine Prozedur, die die Dateisysteme "/", "/stand" und "/usr" mit "**cpio**" in eine große Banddatei sichert.

b) Entwerfen Sie eine Prozedur, die die Dateisysteme "/" und "/stand" mit "**cpio**" in eine kleine Banddatei sichert und "/usr" in die 2. Banddatei.

Lösungsvorschlag zu b):

Was passiert, wenn umountall nicht funktioniert und sich z.B. /home nicht abmontieren läßt?

Aufgaben

Aufgabe 6:

Für **"mount"**, **"umount"**, **"chown"** stehen Ihnen spezielle Kommandos zur Verfügung, die Ihnen gezielt "Super-User"-Rechte verleihen. **Rufen Sie daher kein Kommando mit einem absoluten Pfadnamen auf.**

Erzeugen Sie sich wieder ein eigenes Dateisystem (wie in Aufgabe 3) und montieren Sie Ihr Dateisystem.

a) Kopieren Sie mit **"cpio"**-Kommando alle Dateien und Dateiverzeichnisse unterhalb von "/home/sv0" in das Verzeichnis, an dem Ihr Dateisystem montiert ist (nicht in Ihr HOME-Dateiverzeichnis !).
Erweitern Sie die Zugriffsrechte Ihrer Dateien um das Schreibrecht.

b) Sichern Sie nun Ihr komplettes Dateisystem mit dem **"cpio"**-Kommando in eine Plattendatei "$HOME/cpio_dat".

c) Sichern Sie Ihr Dateisystem zusätzlich mit dem **"tar"** in die Datei "$HOME/tar_dat".

d) Löschen Sie die Dateien in Ihrem montierten Dateisystem (nicht die in Ihrem HOME-Verzeichnis !).
Restaurieren Sie Ihre Daten mit **"cpio"** aus der Datei "$HOME/cpio_dat".

Zusatzaufgabe:

e) Montieren Sie Ihr Dateisystem ab. **Wenn Sie sicher sind, daß Ihr Dateisystem nicht mehr montiert ist,** löschen Sie es mit **"mkfs"**.

Montieren Sie das leere Dateisystem erneut, ändern Sie den Eigentümer und restaurieren Sie Ihre Daten mit **"cpio"** aus Ihrer Sicherungsdatei.

f) Der Kursleiter legt eine 3 1/2-Zoll-Diskette in das Laufwerk.
Versuchen Sie, Daten von dieser Diskette einzulesen.
Welches Format hat die Diskette?

Aufgabe 7:

Löschen Sie einige Dateien in Ihrem Dateisystem.

Restaurieren Sie **nur** die gelöschten Dateien aus Ihrer Sicherungsdatei "cpio_dat" und anschließend noch einmal mit "**tar**" aus der "tar_dat".

Aufgabe 8 (Streamer):

Bitte probieren Sie Ihre Lösung erst aus, wenn Sie vom Kursleiter die Magnetbandkassette erhalten und eingelegt haben.

Schreiben Sie folgende Prozedur:

- Spulen Sie das Band zurück, vermeiden Sie Nachspannen

- `echo "Hallo SINIX" `date` >/dev/tapen` # 0. Banddatei

- Sichern Sie alle Dateien Ihres HOME-Dateiverzeichnisses in der nächsten Banddatei, ohne das Band zurückzuspulen. (1. Banddatei).

- Sichern Sie den Inhalt von "/home/sv0" mit relativen Pfadnamen in der nächsten Banddatei:

- Spulen Sie das Band zurück

- Lesen Sie die Banddateien 0 und 2 in einem Unterdateiverzeichnis von Ihrem HOME-DVZ ein.

Aufgabe 9:

Sie wollen erreichen, daß an Ihrem Rechner nachts um 23:00 Uhr automatisch eine "tar"-Sicherung von "/home"-durchgeführt wird.

Wie stoßen Sie diese Sicherung an ?

Sicherung ganzer Dateisysteme

Das "ufsdump"-Kommando:

- Absolut- oder Differenzsicherungen ganzer **"ufs"-Dateisysteme** (z.B. "/home", "/home1", ...).
- sichert auch Gerätedateien und läßt Zugriffszeiten und Zugriffsrechte unverändert (wie eine physikalische Sicherung)
  —> selbst das Datum des letzten Lesezugriffs bleibt unverändert
- sichert auch bei der Absolutsicherung nur die tatsächlich belegten Blöcke des Dateisystems
- erlaubt verschiedene Stufen von Differenzsicherungen ("Dump-Level": 0 bis 9)
- **sollte nur auf abmontierte und überprüfte Dateisysteme angewendet werden** (sinnvollerweise im Einbenutzer-Betrieb)

  —> andernfalls ist die Sicherung eventuell nicht wieder einlesbar
- ist auch zur Datensicherung auf Platte verwendbar
- "ufsdump"-Sicherungen können von "ufsrestore" wieder eingelesen werden

Die Sicherungsstufen von ufsdump

Stufe 0: Absolutsicherung eines Dateisystems

Stufe n: "ufsdump" sichert alle Dateien des Dateisystems, deren Modifikations-Datum neuer ist als das Datum der letzten notierten Sicherung mit einer Sicherungsstufe < n.

Wird das "ufsdump"-Kommando mit dem Schalter "-u" aufgerufen, hinterlegt es folgende Informationen in der Datei **/etc/dumpdates :**

- Plattenbereich, der gesichert wurde
- Datum und Sicherungsstufe der Sicherung

ufsdump / ufsrestore

Die Kommandos "ufsdump" und "ufsrestore" funktionieren im wesentlichen wie "dump" und "restore" in SINIX V5.2x oder in anderen Berkeley-Systemen.

Unterschiede zu SINIX V5.2x:

- die Kommandos stehen im "usr"-Dateisystem:

  /usr/sbin/ufsdump —> /usr/lib/fs/ufs/ufsdump
  /usr/sbin/ufsrestore —> /usr/lib/fs/ufs/ufsrestore

  Sie sind daher nicht geeignet, das "usr"-System zu restaurieren !

- vor dem 1. Aufruf von "ufsdump" müssen Sie eventuell die Datei **"/etc/dumpdates"** einrichten:

```
>> /etc/dumpdates
```

Beispiel für "ufsdump" auf /home:

```
# als "root" anmelden !!!
cd /
SLICE=c0d0s4  # "slice" fuer /home (siehe /etc/vfstab)

# In Einbenutzer-Betrieb wechseln, alle Prozesse
# beenden, Dateisysteme abhängen:

init 1              # in Einbenutzer-Betrieb herunterfahren
sleep 60
killall 15          # Prozesse geordnet beenden
sleep 10
killall             # restliche Prozesse brutal beenden
sleep 5
umountall           # alles abmontieren (auch /home)
mount /usr          # "ufsdump" steht im "/usr"

fsck -y /dev/rdsk/$SLICE || exit 1
>> /etc/dumpdates

ufsdump -0usf 9000 /dev/tape /dev/rdsk/$SLICE
```

für 150 MB Streamer

Sicherungsstufe: 0 Gesamtsicherung

Beispiel für "ufsrestore" auf /home:

Bevor Sie mit "ufsrestore -r" ein Dateisystem einlesen können, müssen Sie mit "mkfs" ein neues leeres Dateisystem erzeugen. Dazu müssen Sie die "length" des betreffenden "slices" kennen:

```
# als "root" anmelden !!!
prtvtoc /dev/rdsk/c0d0s0    # liefert "length"

# von "Hand" die Größe von slice 4 lesen
...
LENGTH=102000        # in diesem Beispiel: Größe des slice 4

cd /
DSYS=/home
SLICE=c0d0s4         # "slice" fuer /home (siehe /etc/vfstab)

# In Einbenutzer-Betrieb wechseln, alle Prozesse
# beenden, Dateisysteme abhängen:

init 1               # in Einbenutzer-Betrieb herunterfahren
sleep 60
killall 15           # Prozesse geordnet beenden
sleep 10
killall              # restliche Prozesse brutal beenden
sleep 5
umountall            # alles abmontieren (auch /home)
mount /usr           # "ufsrestore" steht im "/usr"

# Dateisystem neu erzeugen und überprüfen:
mkfs -F ufs /dev/rdsk/$SLICE $LENGTH 34 15 || exit 1
#                                  Werte für MC1558 !!!
fsck -F ufs -y /dev/rdsk/$SLICE || exit 1

mount $DSYS          # frisches Dateisystem montieren
                     # (muß in /etc/vfstab stehen)
cd $DSYS
ufsrestore -rf /dev/tape
```

Interaktives "ufsrestore"-Kommando

```
cd ????      # zuerst in das richtige DVZ wechseln
```

`ufsrestore ivf /dev/tape`

* liest die Indexeinträge der auf dem Band mit "ufsdump" gesicherten Dateien ("Inhaltsverzeichnis")

* stellt eine Shell-ähnliche interaktive Schnittstelle zur Verfügung (Prompt "restore> ")

* Sie können sich den Dateibaum auf dem Band ansehen und die Dateien auswählen, die "restore" einlesen soll

Mögliche Kommandos :

- ls
- cd
- pwd

- add [DATEI] a [DATEI]	DATEI (oder Dateiverzeichnis) zum späteren Einlesen auswählen —> in Extraktionsliste eintragen
- delete [DATEI] d [DATEI]	DATEI (oder DVZ) wieder aus der Extraktionsliste löschen

- help oder ?
- quit

- **extract**	**alle Dateien der Extraktionsliste von dem Band oder den Bändern einlesen**

Beispiel für ein Band:

```
restore> extract
specify next volume #: 1
set owner/mode for '.' [yn]: y
```

Hinweis zu "ls" :

"ls" kennzeichnet alle Dateien, die bereits in der Extraktionsliste, stehen mit einem "*".

MINISYSTEM fuer MX300-60 (INTEL)

- Diskette SINIX0 einlegen
- Rechner einschalten
- Disketten SINIX1 und SINIX2 bei Abfrage einlegen
- SINIX2 Abfrage Tastatur --> n eingeben und ENTER
- Naechste Abfrage: continue y/n --> DEL - Taste druecken
- /etc/fs/ufs/fsck /dev/dsk/c0d0s1 (Fragen immer mit y beantworten)
- /etc/fs/ufs/mount /dev/dsk/c0d0s1 /mnt
- Wechsel auf Platte ueber /mnt/... moeglich
- umount /mnt
- uadmin 2 0
- Rechner fährt herunter --> Diskette aus dem Laufwerk entnehmen
- Neu einschalten

Achtung: Beim Test mit automatisch ausfuehrenden Dateien in /etc/inittab
den initdefault auf 1 (SINGLE USER MODE) setzen)

Minisystem fuer MX300-60 (INTEL)

- Diskette SINIX0 einlegen
- Rechner einschalten
- Disketten SINIX1 und SINIX2 bei Abfrage einlegen
- SINIX2 Abfrage Tastatur ----) n eingeben und ENTER
- Naechste Abfrage: continue y/n ----) DEL - Taste druecken
- /etc/fs/ufs/fsck /dev/dsk/c0d0s1 (Fragen immer mit y beantworten)
- /etc/fs/ufs/mount /dev/dsk/c0d0s1 /mnt
- Wechsel auf Platte ueber/mnt/... moeglich
- umount/mnt
- uadmin 2 0
- Rechner faehrt herunter --) Diskette aus dem Laufwerk entnehmen
- Neu einschalten

Achtung: Beim Test mit automatisch ausfuehrenden Dateien in /etc/inittab den initdefault auf 1 (SINGLE USER MODE) setzen.

Installation, Notreparatur, Konfiguration, Spoolsystem

Key-Diskette

Bei neuen MX300 Rechnern ist das Betriebssystem bereits ab Werk auf der Platte vorinstalliert.

Sie müssen das Betriebssystem beim ersten Systemstart mit der mitgelieferten "KEY"-Diskette ("K.DISK", "AUTHORIZATION FILE") aktivieren.

Wenn Sie den Rechner zum ersten mal einschalten, erhalten Sie nach den normalen Meldungen beim Systemstart die Aufforderung:

```
Please install your SINIX authorization file
Insert diskette into Floppy Drive 1.
Type [go] when ready,
or [q] to quit:
```

Legen Sie Ihre Key-Diskette ein und geben Sie ein: **go**

Nach der Installation erhalten Sie die gleiche Aufforderung noch einmal:

```
Insert diskette into Floppy Drive 1.
Type [go] when ready,
or [q] to quit:
```

Geben Sie nun ein: **q**

Anschließend müssen Sie eventuell noch ein Korrekturband oder Korrekturdisketten installieren.
Den Umfang des vorinstallierten Betriebssystems entnehmen Sie Ihrer SINIX-Freigabemitteilung.

Vorinstalliertes Betriebssystem

MX300-Rechner werden normalerweise bereits im Werk auf der 1. Platte vorinstalliert.

Sie müssen **"nur"** noch (in dieser Reihenfolge):

- beim ersten Systemstart die Key-Diskette **"K.Disk"**(**"SINIX authorization file"**) installieren. Ohne Key-Diskette fährt das System nicht hoch !
- ev. ein aktuelles Korrekturband **"SINIXK"** installieren
- Bildschirmarbeitsplätze und Drucker konfigurieren
- nicht benötigte Pakete deinstallieren: pkgrm ...
- eventuell die 2. Platte aktivieren
  Aus den Meldungen beim Systemstart erkennen Sie, welche Platten in Ihren Rechner eingebaut sind (Plattentyp !).
- eventuell den Rechnernamen ändern

  Der Rechner- und der Domain-Name ist beim vorinstallierten System auf **"SNI"** gesetzt.
- Kennwörter ändern:
  Die Benutzer **"root"**, **"admin"**, **"tele"**, **"sysadm"** und **"install"** haben beim vorinstallierten System das Kennwort: "SNI"
- Anwendersoftware installieren
- Benutzergruppen und Benutzerkennungen einrichten

Rechnernamen ändern:

Es gibt kein Kommando **"hostname"** !

1) Rechner vom LAN abhängen (**"admin"**-Menü oder **"/etc/default/inet"** ändern und **"init 6"**)

2) `sh /etc/init.d/inetinit down` # Prozesse beenden

3) Rechnernamen über "sysadm" ändern: "node name"
   /etc/net/ticots/hosts, ../ticlts/hosts, ../ticotsord/hosts
   Achtung: "System name" muß "SINIX-L" bzw. "SINIX-M" bleiben.

4) `uname -S RECHNERNAME` # Rechnername einstellen

5) `init 6` # Reboot

sysadm in V5.4

UNIX System V.4 bietet dem Systemverwalter ein fenster-orientiertes Menüsystem "sysadm" (OA&M - Menüs) in englischer Sprache. Die Menüdateien finden Sie unterhalb von "/usr/sadm/sysadm/menu".

Sie können das "sysadm"-Menü aufrufen:

- aus der "root"-Kennung mit dem **Kommando : "sysadm"**

- direkt über die spezielle **Benutzerkennung: "sysadm"**

"sysadm" bietet Ihnen folgendes Hauptmenü:

UNIX System V/386 Operations, Administration and Maintenance
1 UNIX System V Administration

applications	Administration for Available Applications
backup_service	Backup Scheduling, Setup and Control
buses	Bus Information and Management
diagnostics	Diagnosing System Errors
file_systems	File System Creation, Checking and Mounting
machine	Machine Configuration, Display and Shutdown
network_services	Network Services Administration
networks	Network Administration
ports	Port Access Services and Monitors
preSVR4	Peripherals Setup
restore_service	Restore From Backup Data
schedule_task	Schedule Automatic Task
software	Software Installation and Removal
storage_devices	Storage Device Operations and Definitions
system_setup	System Name, Date/Time and Initial Password Setup
users	User Login and Group Administration

In SINIX V5.4 sind mehrere Menüpunkte "abgeklemmt"(z.B. "users" oder "backup"), da Sie dafür das "admin"-Bediensystem (COLLAGE) oder Kommandos verwenden sollen.

Sie können "sysadm" bereits mit einer Menüauswahl als Argument aufrufen:

sysadm system_setup # Rechnernamen: "node name" ändern
sysadm networks

Installation von Softwareprodukten

Sie wollen zusätzliche SINIX-Softwareprodukte installieren, beispielsweise:

HIT, SIPLAN, INFORMIX, COLLAGE, OCIS, SIDRAW,
SICHART, ONLINE-Dokumentation, CCP-STA?, ...

Lesen Sie für **jedes Softwareprodukt** die produktspezifischen Installationshinweise im Benutzerhandbuch oder in der Freigabemitteilung des Softwareprodukts bevor Sie mit der Installation beginnen.

Softwareprodukte können Sie entweder über das **"admin"**-Bediensystem oder über das Kommando "pkgadd" installieren.
Die Installation, Deinstallation und Verwaltung von Softwareprodukten erfolgt in V5.4 über die Kommandos:

pkgadd Installation von SINIX/UNIX 5.4 Softwarepaketen

```
/usr/sbin/pkgadd -d ctape1        # vom Streamer

/usr/sbin/pkgadd -d diskette1     # 3,5 Zoll Diskette
```

Mit **"pkgadd"** können Sie nur Softwareprodukte installieren, die im **"package"**-Format vorliegen (neu in UNIX System V.4).
Softwarepakete für System V.3/386 müssen Sie mit dem **"custom"**-Kommando installieren.

pkginfo Informationen über installierte Softwareprodukte ausgeben:

```
pkginfo | pg                      # Übersicht über alle Pakete
pkginfo -l SIces                  # alle Informationen zu SIces
```

pkgrm Deinstallation von Softwarepaketen

```
/usr/sbin/pkgrm                   # Menüauswahl
```

Sie erhalten eine Übersicht der installierten Pakete und können menügeführt die Pakete auswählen, die Sie deinstallieren wollen.

```
/usr/sbin/pkgrm rfs lp  SIsim     # Pakete deinstallieren
```

Software-Pakete

SINIX V5.4 enthält neben dem Basissystem zahlreiche Softwarepakete.

Bei allen Rechnern, mit Ausnahme des MX300-45, wird empfohlen, alle Pakete zu installieren.
Sie können nachträglich einige Pakete wieder deinstallieren (z.B. "rfs" oder "lp").

Pakete, die SINIX-Erweiterungen gegenüber dem AT&T UNIX-System V.4 enthalten, haben einen Namen, der mit "SI" beginnt.

Kurzbeschreibung der Pakete bzw. Paketgruppen:

SIcoldrv SIcolrts SIcolface	COLLAGE-Laufzeitsystem und Bediensystem (muß installiert werden)
SIFonts SIMetapr	Druckprogramm für COLLAGE und COLLAGE- Anwendungen
SImshrs SImesui	Menü-Laufzeitsystem (MES) Menü-Bediensystem
SIlpr	SINIX Spoolsystem V3.0
lp	Alternative zu SIlpr: das AT&T Spoolsystem (nicht installieren)
SIcompat	Kommandos aus SINIX V5.2x (muß installiert werden)
terminf ed	terminfo-Paket: Sie benötigen "siemens.ti" Editor Paket: **"vi"**, **"spell"**, ... muß beides installiert werden)

wird z.B. benötigt zur Administration von LAN

Pakete zur Netzanbindung (TCP/IP, LAN1):

nsu inet rpc	Basispakete zur Netzanbindung und zur Interprozeßkommunikation über "sockets"
SIexos201	SINIX Treiber für Ethernet Baugruppe
SIlanadm	SINIX LAN Administration (Menü)

Zusätzlich (bei Bedarf):

nfs dfs	Verteiltes Dateisystem (NFS bzw. DFS) **"Network File System"**
rfs	Alternative zu NFS: **"Remote File Sharing"**

Software-Pakete (Fortsetzung)

C-Entwicklungssystem (nur für Programmierer):
(fehlt beim MX300-45)

scde	Standard C Entwicklungssystem von AT&T
SIces	SINIX Erweiterungen zum CES
SICEScp	Interaktives Programmiersystem auf Basis
SIcp	von COLLAGE: ColProg

Unterstützung von SINIX-Besonderheiten: (nicht MX300-45)

SIsim	Terminalanschluß über TACSI (SIM-Baugruppe)
SIscsi	SCSI-Unterstützung für Exabyte, Wormlaufwerke, externe SCSI-Platten,
...	
SIvpss	VPSS Spiegelplatten Unterstützung

Für das AT&T Menüsystem (z.B. "sysadm") benötigen Sie:

fmli	AT&T Form and Menu Language Interpreter
oam	Operations, Administration and Maintenance (es genügt, wenn Sie das **"BASIC user interface"** installieren)

Weitere Pakete:

xcp	XENIX-Kommandos, MS-DOS Disketten bearbeiten, XENIX-Gerätedateien für Disketten	
compat	Berkeley-Kommandos und Bibliotheken (/usr/ucb/*)	
termcap	termcap-Kompatibilität	
face	AT&T Framed Access Command Environment	nicht notwendig
windowing	AT&T Fenstersystem	

Die Installation der Pakete erfolgt in zufälliger Reihenfolge.

Sie können jedes Paket nachträglich installieren oder deinstallieren:

```
/usr/sbin/pkgadd -d /tape1        # von SINIX3 installieren
                                  # dauert sehr lange !
```

Erzeugen von Softwarepaketen: (für Entwickler)

2 Formate für "packages"

- Dateibaum-Struktur — auf Platte / s5-Dateisystem auf Floppy
- "datastream"-Format für beliebige Datenträger

"package" erzeugen:
pkgmk-Kommando: erzeugt Dateibaumstruktur

"package" umsetzen (z.B. in "datastream"):
pkgtrans-Kommando

Informationsdateien für pkgmk:

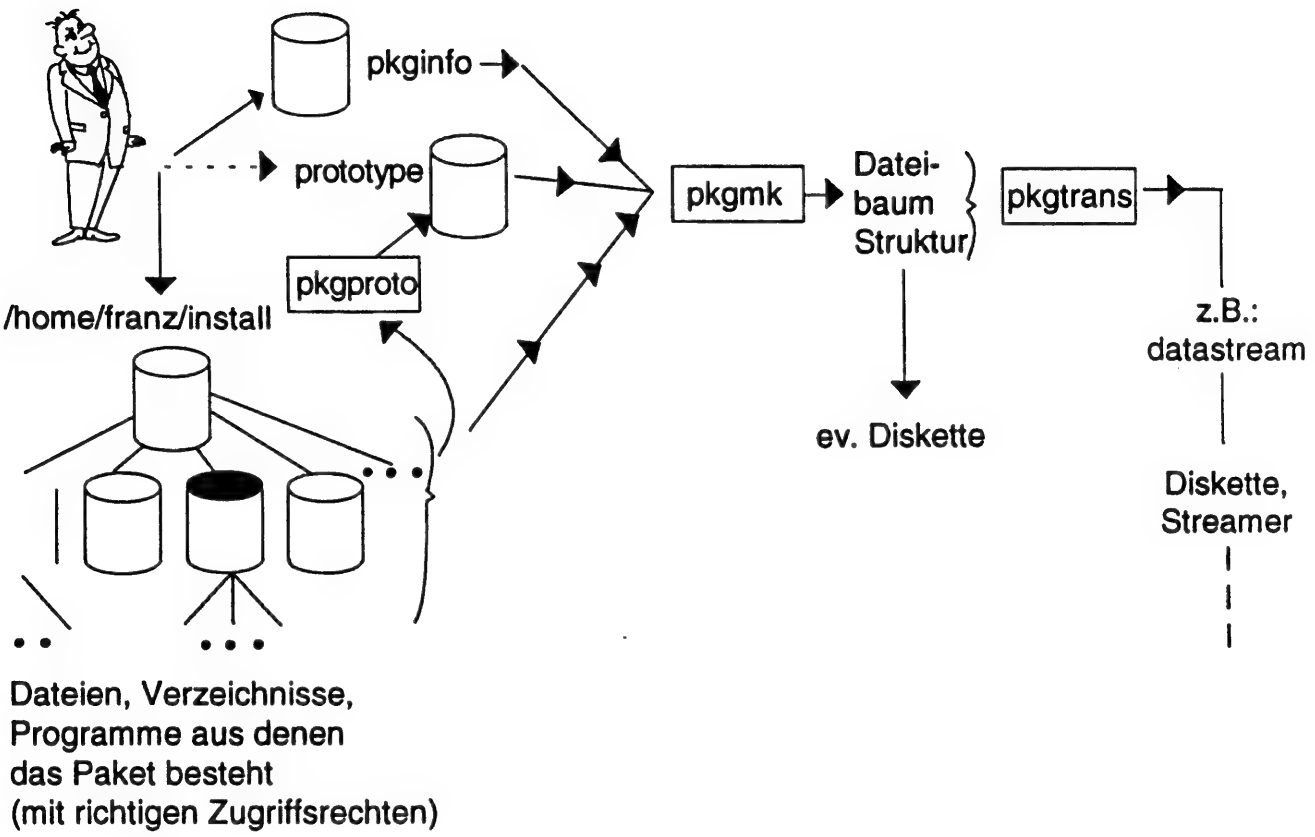
pkginfo-Datei (müssen Sie erstellen)
prototype-Datei: Liste des gesamten Pakets mit allen Informationen

Hilfsprogramm zum Erzeugen der prototype-Datei:
echo "i pgkinfo" > prototype
pkgproto /home/franz/install=/opt >>prototype

unterhalb dieses Pfadnamens muß sich ein genaues Abbild des Pakets befinden.

Paket auf Diskette erzeugen:

pkgmk -d /dev/rdsk/f03ht mon


pkginfo
prototype
pkgproto
/home/franz/install
pkgmk
Datei-
baum
Struktur
ev. Diskette
pkgtrans
z.B.:
datastream
Diskette,
Streamer
Dateien, Verzeichnisse,
Programme aus denen
das Paket besteht
(mit richtigen Zugriffsrechten)

Datei: pkginfo siehe pkginfo (4)

* ***PGK***=*mon*	Package-Name
* ***NAME***=*Software Monitor*	
* ***CLASSES***=*none*	muß übereinstimmen mit "Class" in
* ***VERSION***=*1*	prototype
* ***CATE GORY***=*applications*	
* ***VENDOR***=*Walter Mecky*	
* ***ARCH***=*i386*	

-
-
-

(*) Pflichtfelder

Literatur:

SINIX V5.4 Systemschnittstellen und Werkzeuge für die Anwendungs-programmierunng

Leitfaden für Programmierer: U6414-J-Z95-1

Kapitel: "Packaging von Anwendersoftware"

Installation auf dem MX300

Installationssatz

- SINIX0, SINIX1, SINIX2 Disketten
- SINIX3 Magnetbandkassette
- K.Disk Key-Diskette: "SINIX authorization file"
- **SINIX** Korrekturband: **SI**update

In der Liefereinheit "**Message-Files**" ist zusätzlich enthalten:

- README Diskette Softwarepaket mit aktueller Freigabemitteilung

Installationsablauf (MX300)

➪ SINIX0 einlegen und Rechner einschalten Aufforderung abwarten und:

➪ SINIX1 einlegen und bestätigen Aufforderung abwarten und:

➪ SINIX2 einlegen und bestätigen

➪ Sie können wählen, ob Sie das System nur auf der 1. Platte oder gleich auf beiden Platten installieren wollen. Wenn Sie das System nur auf der 1. Platte installieren, können Sie die 2. Platte nachträglich aktivieren (und beliebig einteilen).

➪ eine UNIX-Partition einrichten (100%, ACTIVE) (wie bei "**fdisk**")

➪ Typ der Dateisysteme "**/**", "**/usr**", "**/home**", "**/var**" und "**/opt**" festlegen ("**ufs**" oder "**s5**").

optional: "**/tmp**" Dateisystem anlegen
Die Installationsprozedur bietet Ihnen nicht an, weitere "**slices**" einzurichten.

➪ Größe der Dateisysteme in Zylindern bzw. in Sektoren festlegen (vgl. SINIX-Freigabemitteilung)

➪ Die Installationsprozedur erzeugt die Dateisysteme auf der Festplatte und installiert ein Minisystem auf der Festplatte.

➪ Das System "**bootet**" von der Festplatte und die weitere Installation erfolgt vom SINIX3-Band (siehe übernächste Seite)

Installation auf dem MX500-90

Installationssatz:

- SINIX0 Magnetbandkassette (155 MB-Streamer-Band)
- K.Disk Key-Diskette "SINIX authorization file"
- ??????? weitere Korrekturbänder oder Disketten

Installationsablauf (MX500):

- SINIX0-Band einlegen, Taste AUTOBOOT ausschalten, Rechner einschalten und auf das * des Hardwaremonitors warten. Falls der Rechner automatisch hochfährt, müssen Sie AUTOBOOT ausschalten und noch einmal beginnen.
- Bootkommando eingeben, um vom Band zu booten:

  b0 ts(32,1) root=ramd(0) swap=ramd(1)

Das root-Dateisystem wird nun automatisch in die RAM-Disk eingelesen.

- AUTOBOOT wieder einschalten, damit die MX500 in Zukunft automatisch hochfahren kann.
- Die Installationsprozedur stellt fest, welche Platten angeschlossen sind. Sie können eine **"primary disk"** und eine **"secondary disk"** für die Installation auswählen.
- Sie werden gefragt, wie Sie "/stand", "swap", "/" "/usr", "/home", "/var", "/opt" und "/tmp" auf diese beiden Platten verteilen wollen. Sie können auch noch weitere Slices für Dateisysteme und für Arbeitsspeicherabzüge ("/dev/dump") konfigurieren.
- Die Installationsprozedur erzeugt die Dateisysteme auf der Festplatte und installiert ein Minisystem auf der Festplatte.
- Das System bootet von der Festplatte. Anschließend wird automatisch das SINIX-Grundsystem (Basissystem) eingespielt (siehe nächste Seite).

Installation vom SINIX3 bzw. SINIX0-Band

Die weitere Installation vom SINIX-Band besteht aus 3 Teilen:

1) Installation eines Basissystems (20 Minuten)

2) Eingabe von Kennwörtern und Netzwerknamen und der Internetadresse

3) Installation der verschiedenen Software-Pakete

⇨ MX300: bis zu 3 1/2 Stunden mit vielen Abfragen
⇨ MX500: ca. 1 Stunde mit wenigen Abfragen

Installationsanleitung:

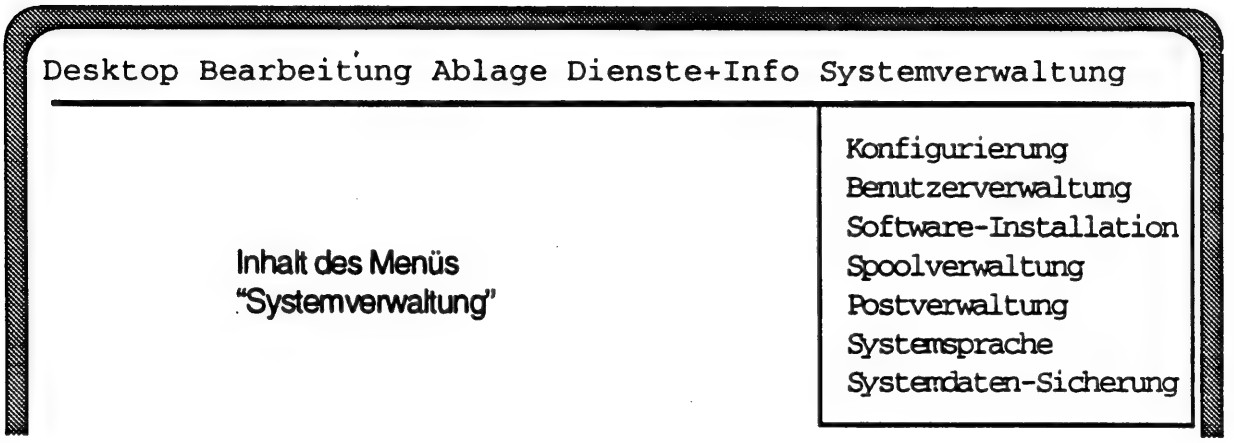
Eine genaue Installationsanleitung finden Sie in der SINIX-Freigabemitteilung.

Platteneinteilung:

Schreiben Sie sich bei der Installation die Größe jedes **"slices"** und den Dateisystemtyp (**"s5"** bzw. **"ufs"**) auf !
Die Größe eines Zylinders hängt vom Plattentyp ab.

Konfigurierung der Peripherie

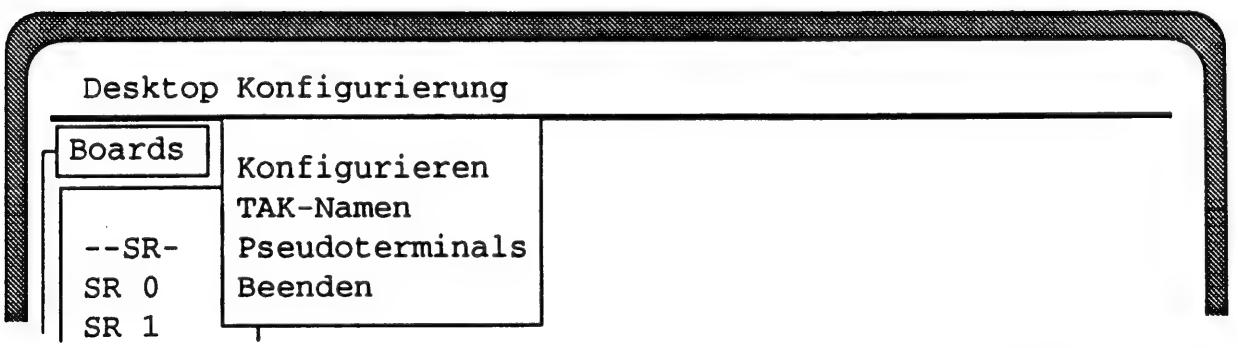
Die Konfigurierung der E/A-Prozessoren für Terminals, Drucker und anderer Peripheriegeräte sollten Sie ausschließlich über das "admin"-Bediensystem durchführen



Sie können das Menü nur auswählen, wenn ein Verzeichnisfenster offen ist und "oben" liegt.

Über den Menüeintrag "Konfigurierung" starten Sie das Programm "**Konf**" des Bediensystems. Dieses Programm dürfen Sie nicht von zwei Terminals gleichzeitig verwenden.

Sie erhalten ein Fenster "Boards", das eine Liste aller vorhandenen E/A-Baugruppen enthält. Außerdem erhalten Sie eine neue Menüleiste mit folgendem Menü:

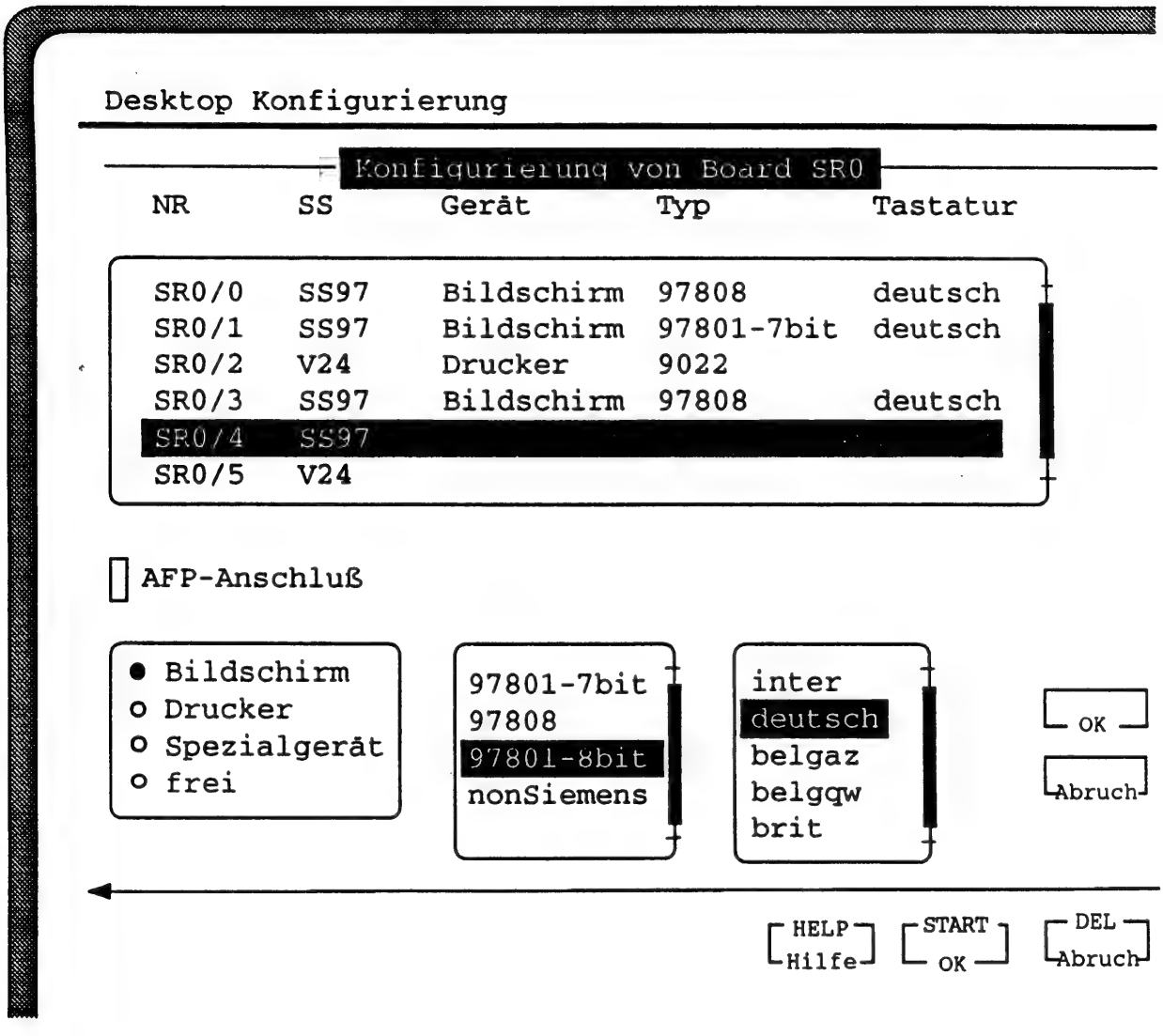


Sie wählen im Fenster "**Boards**" eine E/A-Baugruppe aus und anschließend aus dem Menü den Eintrag"**Konfigurieren**".

Das Bediensystem zeigt Ihnen die aktuelle Konfigurierung der betreffenden E/A-Baugruppe. Sie wählen eine oder mehrere Schnittstellen aus, die Sie umkonfigurieren wollen.

Auswahlmöglichkeiten:

Das Bediensystem gibt Ihnen nun zahlreiche Auswahlmöglichkeiten für diese Schnittstelle:



Haben Sie Ihre Auswahl für diese Schnittstelle beendet, so betätigen Sie die kleine OK-Marke.

Haben Sie Ihre Auswahl für die gesamte E/A-Baugruppe fertig, dann drücken Sie die START-Taste.

Systemdateien

Systemdateien der verschiedenen UNIX/SINIX Versionen:

	UNIX V.40	SINIX 5.4/SINIX-ODT UNIX V.3	SINIX 5.2x Berkeley-UNIX
Konfigurationsdatei für Terminals	Kommando: "sacadm"	/etc/inittab	/etc/ttys
Programm zur Terminalverwaltung	ttymon	getty	getty
Schnittstellen-parameter	/etc/ttydefs Kommando: "sttydefs"	/etc/gettydefs	/etc/gettytab
TERM-Variable (Terminaltyp)		/etc/ttytype	
Terminal initialisierung		/etc/termtab is=...:LK..=...	/etc/termcap is=...:LK=..
Eröffnungs bildschirm	/etc/issue		/etc/herald/*

Die Datei /etc/ttytype:

*97801 pts** —— Pseudoterminals
97801 term/tty000 —— Standardterminal
97808 term/tty003 —— Grafik-COLLAGE-Terminal

TERM-Variable

Terminalname (Schnittstelle)

Die Datei /etc/termtab (nur SINIX V5.4):

term/tty000:is=\E[21u\08 \08\E(K\E[7u:LK=deut
term/tty003:is=\E[21u\08 \08\E(B\E[6u:LK=int

Terminalname

Tastaturtabelle für /sbin/keyload

Steuerzeichen zur Initialisierung

gettydefs oder ttydefs

UNIX V.4 Systeme, die mit "getty"-Prozessen arbeiten (z.B. SINIX V5.4), benötigen die Datei "/etc/gettydefs":

Auszüge aus der /etc/gettydefs von SINIX V5.4:

```
S# B38400 OPOST ONLCR ... ISTRIP ... CS7 CREAD PARENB PARODD # B38400 OPOST ... PARODD #login: #S

N# B38400 OPOST ONLCR ... CS8 CREAD # B38400 OPOST ... CREAD #login: #N

B# B38400 OPOST ONLCR ... ISTRIP ... CS7 CREAD PARENB PARODD # B38400 OPOST ... PARODD  #login: #B

V# B19200 OPOST  ONLCR ... CS8 CREAD # B19200 OPOST ... CREAD # login: #v
```

Label der Zeile, die als nächstes verwendet wird

Treibereinstellungen (siehe "stty"-Kmdo)

Kennzeichnung/Label der Zeile (siehe/etc/inittab)

Initialisierungs-einstellung

endgültigeEinstellung

Aufforderungstext zur Eingabe der Benutzerkennung

UNIX V.4. Systeme, die mit "ttymon" arbeiten, benötigen die Datei "/etc/ttydefs":

Auszüge aus einer /etc/ttydefs:

```
S: 38400 opost  onlcr ... istrip ... cs7 cread parenb parodd : 38400 opost ... parodd : : S

N: 38400 opost  onlcr ... cs8 cread : 38400 opost ... cread : : N

B: 38400 opost  onlcr ... istrip ... cs7 cread parenb parodd : 38400 opost  ... parodd : : B

v: 19200 opost  onlcr ... cs8 cread : 19200 opost ... cread : : v

auto:hupl:ixany sane tab3 hupl:A:9600
```

"autobaud"

Die Datei "/etc/ttydefs" sollten Sie nur mit dem Kommando **"sttydefs"** bearbeiten.

terminfo statt termcap

In früheren UNIX-Versionen holten sich die Anwenderprogramme alle Steuerzeichen und Informationen zur Terminalsteuerung aus einer großen Datei "/etc/termcap".

Die neuen UNIX-Versionen stellen den Anwenderprogrammen kleine binäre "terminfo"-Beschreibungsdateien zur Verfügung.

In SINIX V5.4 wird die "/etc/termcap" nur noch für alte Anwenderprogramme benötigt, alle neuen Programme arbeiten unterhalb des Dateiverzeichnisses:

/usr/share/lib/terminfo

Die Beschreibungsdatei für das 97801-Terminal hat beispielsweise den Pfadnamen:

/usr/share/lib/terminfo/9/97801

Die binären Dateien können Sie mit dem Kommando **"infocmp"** in Quelldateien zurückverwandeln:

infocmp 97801 > 97801.ti

Sie können die Quelldatei verändern und mit dem **terminfo compiler (tic)** wieder neu übersetzen:

tic 97801.ti

"tic" legt das Ergebnis normalerweise wieder unter "/usr/share/lib/terminfo/?/..."ab.

TERMINFO

Über die Umgebungsvariable "TERMINFO" können Sie ein anderes Dateiverzeichnis für die terminfo Dateien einstellen. Falls diese Umgebungsvariable definiert ist, wird der Inhalt anstelle von "/usr/share/lib/terminfo" verwendet, um die binäre Datei zu finden oder zu erzeugen. Der Dateiname lautet dann:

$TERMINFO/?/ $TERM

Eigenschaften von "terminfo"

- Verbindliche Festlegung der Eigenschaften und ihrer Feldnamen (ähnlich ANSI X3.64-1979)
- Komma als Feldtrenner, Leerzeichen zwischen Feldern erlaubt
- Felder eines Eintrags sind durch führende Blank(s) oder Tabulator gekennzeichnet.
- die leistungsfähige "curses"-Bibliothek für bildschirm-orientierte Anwendungen arbeitet mit "terminfo"
- Shellkommando zur Ausgabe von Steuerzeichen aus der terminfo-Datei:

  tput clear oder *US= `**tput** smul`*

Feldnamen bei "terminfo":

terminf	Bedeutung
cols/lines	Anz. Spalten/Zeilen des Bildschirms
cup	allgemeine Cursorpositionierung
cuf1/cub1	Cursor nach rechts / links
cud1/cuu1	Cursor nach unten / oben
el	Löschen: Cursor bis Zeilenende
ed	Löschen: Cursor bis Bildschirmende
clear	**Bildschirm löschen**
dl1/il1	Zeile löschen / einfügen
smcup	Terminalinitialisierung
smul/rmul	Unterstreichen ein / aus
kcud1 / kcuul1	Tasten "cursor down / up"
kcub1 / kcuf1	Tasten "cursor back / forward"
kich1 / kdch1	Tasten "INS CHAR / DEL CHAR
kf1 - kf20	Funktionstasten F1 - F20
kf23 - kf44	<Shift F1> bis <Shift F20>
lf0 - lf"	HELP, START, END
khlp, kress	HELP, START

Die SINIX Druckerverwaltung

lpr /opt/bin/lpr
Benutzerschnittstelle der Druckerverwaltung (wird von Benutzern oder von Programmen aufgerufen)

daemon /opt/etc/daemon
eigentliche Druckerverwaltung (2 Hintergrundprozesse)

/opt/etc/startup
Shellprozedur zum Start der Druckerverwaltung

/var/spool/spooler/tmp/daemtrc
Fehlermeldungsdatei

/var/spool/spooler/CONFIG

Textversion der Konfigurationsdatei:

Diese Datei enthält alle konfigurierten Drucker, Druckergruppen, Druckerverwalter und die Zuordnung von Terminals oder Benutzern zu bestimmten Druckergruppen.
Die CONFIG wird bei der Konfiguration vom **"admin"**- Bediensystem angelegt und kann vom Systemverwalter modifiziert werden.

/var/spool/spooler/FORMTAB

Das SINIX Spoolsystem erlaubt mehrere Warteschlangen je Drucker zur Unterstützung unterschiedlicher Papierformulare. In der FORMTAB Datei sind die erlaubten Formularnummern und Formularnamen hinterlegt.

/var/spool/spooler/sp/sp.*

Ruft der Anwender den **"lpr"** über Pipe oder mit einem der Schalter **"-cp"** oder **"+co"** auf, so kopiert der **"lpr"** die zu druckende Datei in dieses Dateiverzeichnis (temporäre Druckdatei).

Die CONFIG-Datei

1. Teil: konfigurierte Drucker

In jeder (langen) Zeile stehen folgende Informationen für einen Drucker (durch Blank getrennt):

- Druckername (max. 5 Zeichen)
- Backendaufruf mit absolutem Pfad (in Apostrophen)
- Absoluter Pfadname der Gerätedatei
- Liste aller zugelassenen druckerspez. lpr-Schalter

2. Teil: Definition der Druckergruppen

3. Teil: Liste der "Druckerverwalter"

4. Teil: Zuordnung von Terminals oder Benutzern zu Druckergruppen

Beispiel für eine CONFIG-Datei (mit verkürzten Zeilen):

```
D009 '/opt/etc/lp9001-b +hd' /dev/term/tty009 -zs= -zb= +tab -mar= -top= -za= -cat +cat -form=
D008 '/opt/etc/lp9001' /dev/term/tty008 -zs= -zb= -font= +tab -mar= -top= -za= -cat +cat -form=
D006 '/opt/etc/lp9004'/dev/term/tty006 -pl= -zs= -pb2 -hop= +tab -mar= -top= -za= -form=
D002 '/opt/etc/interface +lkmod -prog=/bin/cat +cbreak +odd -speed=B9600' /dev/term/tty002

9001 ( D008 D009 ) 'nur 9001 Drucker D008 D009 '
G009 ( D009 ) 'DRUCKER 9001-b'
G008 ( D008 ) 'DRUCKER 9001'
G006 ( D006 ) 'Hier muss ein Kommentar stehen !'
G002 ( D002 ) 'DRUCKER 9plott'

admin ( D009 D008 D006 D002 )
franz ( D009 )
meier ( D002 )

/dev/term/tty000 G009
/dev/term/tty001 G002
/dev/term/tty004 G008
/dev/term/tty007 G006
```

Die 4 Teile der CONFIG-Datei sind durch jeweils eine Leerzeile getrennt.
Zusätzliche Leerzeilen dürfen nicht vorkommen.

Neustart des Spoolsystems mit neuer CONFIG :

Wenn eine Veränderung der CONFIG-Datei wirksam werden soll, muß der Systemverwalter die Druckerverwaltung neu starten. Dies geschieht über das Kommando:

lpr -rr

Falls die Druckerverwaltung nicht mehr aktiv ist, können Sie sie starten mit der Startupprozedur:

/opt/etc/startup

Alle Meldungen, die die Druckerverwaltung beim Start abgibt, stehen anschließend in der "daemtrc"-Datei :

/var/spool/spooler/tmp/daemtrc

Falls die CONFIG-Datei Syntaxfehler enthält, stehen entsprechende Fehlermeldungen in der **"daemtrc"**-Datei.
In diesem Fall startet das Spoolsystem nicht und
alle Drucker sind **"UNBEKANNT"**.

Es ist wichtig, daß alle Backends und Gerätedateien, die in der CONFIG-Datei angegeben sind, auch wirklich existieren und ansprechbar sind !
Andernfalls bleiben die betreffenden Drucker im Zustand **"UNBEKANNT"**.

Legen Sie sicherheitshalber vor jeder Änderung der CONFIG ein Sicherheitskopie dieser Datei an !

Notreparatur

Angenommen, Ihre "**/**" oder "**/stand**" Dateisysteme sind so defekt, daß Ihr Rechner nicht mehr hochfährt.

In diesem Fall bleiben Ihnen nur die Boot-Disketten und Ihre (hoffentlich vorhandene) "**cpio**"-Sicherung von "**/**" und "**/stand**". Glücklicherweise befindet sich auf der "SINIX2"-Diskette das Kommando "**cpio**".

Falls Plattenfehler die Ursache für Ihr Problem sind, müssen Sie zunächst die Plattenfehler beseitigen lassen (TDS-Diskette) oder die Platte austauschen, bevor Sie die Dateisysteme restaurieren.

Booten von Diskette oder vom SINIX0-Band:

- Beginnen Sie die Installation, wie auf den Seiten L 5-10 bzw. L 5-11 beschrieben.
- Booten von SINIX0, für Minirootsystem und swap-Bereich werden RAM-Disks verwendet.
  Beim MX300 wird die SINIX2-Diskette unter /install an die RAM-Disk montiert.
- Brechen Sie die Installation bei der ersten echten Abfrage mit <DEL> ab.
- Sie befinden sich nun auf der RAM-Disk.
  Beim MX300 dürfen Sie die montierte SINIX2 nicht entnehmen.

Wir müssen 2 Fälle unterscheiden:

A) Ihre 1. Platte ist noch in Ordnung oder repariert, die Platteneinteilung ist unverändert, so daß nur die Dateisysteme überprüft (eventuell neu erzeugt) und die fehlenden Daten eingespielt werden müssen.

B) Ihre 1. Platte wurde ausgewechselt oder Sie wollen die Platteneinteilung verändern.
In diesem Fall rufen Sie auf:

```
fdisk /dev/rdsk/c0d0s0                  # UNIX-Partition einrichten

disksetup -IB /dev/rdsk/c0d0s0          # "slices" einrichten
                                        # Dateisysteme erzeugen
```

Dateisysteme prüfen (erzeugen) und Daten einspielen:

Auf dem RAM-Disk-System haben Sie nur wenig Kommandos zur Verfügung!
Anstelle von "**ls**" können Sie "**echo ***" verwenden.

Falls "*/*"-Dateisystem auf der Platte vom Typ "**ufs**" ist:

```
# eventuell Dateisystem neu einrichten ?
# /etc/fs/ufs/mkfs /dev/rroot LENGTH NSECT NTRACK ?

/etc/fs/ufs/fsck -y /dev/rroot          # prüfen

/etc/fs/ufs/mount /dev/root /mnt        # montieren
```

Das Dateisystem "**/stand**" (mit Systemkern):

```
# eventuell Dateisystem neu einrichten ?
# /etc/fs/bfs/mkfs /dev/rdsk/c0d0s10 LENGTH ?

/etc/fs/bfs/fsck /dev/rdsk/c0d0s10      # prüfen

mkdir /mnt/stand                        # sollte bereits vorhanden sein !

/etc/fs/bfs/mount /dev/dsk/c0d0s10 /mnt/stand
```

Wie prüfen und montieren wir das **"/usr"**-Dateisystem ?

PATH erweitern, damit Sie Kommandos der Platte zur Verfügung haben:

```
PATH=$PATH:/mnt/usr/bin:/mnt/usr/sbin:
export PATH
```

"cpio"-Sicherungen einlesen:

Die **"cpio"**-Sicherungen müssen mit relativem Pfadnamen gesichert sein !!!!!

```
cd /mnt                          # "/"-Bereich der Platte
umask 0

cpio -ivBdmu < /dev/rmt/c0s0     # überschreiben
                                 # im Minisystem existiert kein "/dev/tape"
```

"cpio" überschreibt normalerweise keine Dateien neueren Datums sondern meldet solche Dateien als **"Fehler"**.
Wenn Sie den Schalter **"-u"** angeben, überschreibt **"cpio"** auch neuere Dateien.

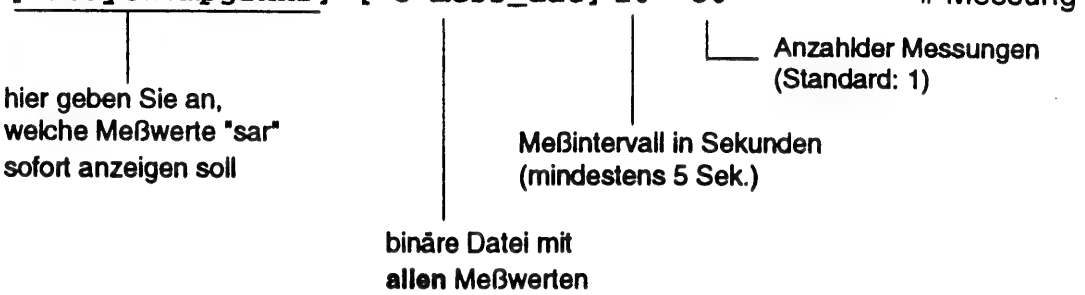
```
cd /
init 6                           # neu hochfahren
```

Systhem - Tuning

sar: system activity reporter

Systemaktivitäten messen, anzeigen und speichern:

```
sar [-ubdycwvmpgrkxD] [-o mess_dat] 10  80          # Messung
```



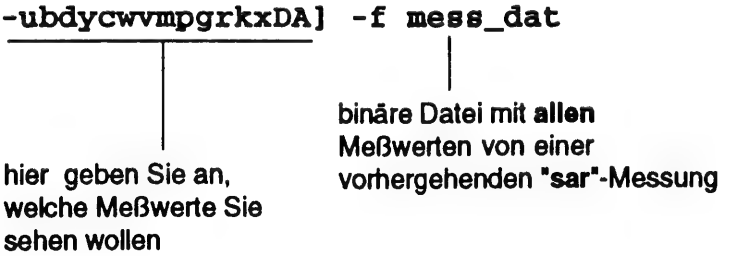
Der UNIX-Kern enthält zahlreiche Zähler, die bei den verschiedensten Systemaktivitäten hochgezählt werden.
"**sar**" liest alle Zähler und speichert sie in der binären Ausgabedatei, die Sie über den Schalter "**-o**" angegeben haben.

Über Schalter können Sie festlegen, welche Systemaktiviäten "**sar**" sofort auf der Standard-Ausgabe ausgibt. Sie sollten zunächst maximal einen Schalter angeben, weil sonst die Ausgabe nicht lesbar ist.

Auswertung der Messdatei mit "sar":

Sie haben mit dem ersten "sar"-Kommando eine binäre Messdatei erzeugt, die Sie nun nachträglich auswerten können:

```
sar   [-ubdycwvmpgrkxDA] -f mess_dat                # Auswertung
```



Bei der Auswertung einer Meßdatei können Sie mehrere Schalter gleichzeitig angeben. Sie erhalten dann mehrere Auswertungstabellen hintereinander.

Bedeutung einiger Ausgabe-Schalter:

-u	CPU-Benutzung (Standard, falls kein Schalter angegeben ist)
-D	D wie **"-u"** aber mit getrennter Berücksichtigungvon fernen (remote) Rechnern im Netz
-d	Plattenaktivitäten
-y	Terminal Ein-/Ausgabe
-c	Systemaufrufe
-cD	wie **"-c"** aber mit getrennter Berücksichtigung von fernen (remote)-Rechnern im Netz
-w	**"swapping"** und Prozeßwechsel
-p	**"paging"**-Aktivitäten
-r	freie 4 KB Seiten im Arbeitsspeicher und freie 4 KB Blöcke im **"swap"**-Bereich
-v	Auslastung von Systemtabellen (Prozeß-Tabelle, Inode-Tabelle, offene Dateien, Locking-Tabelle)

Die Größenangabe SZ bei **"ps -el"** erfolgt ebenfalls in 4 KB Seiten !

Beispiele für sar

Messung:

```
sar -o mess_dat 5 10          # 10 Messungen mit 5 Sekunden-
                              # Intervall durchführen
```

Sie sollten das Messintervall normalerweise größer als 5 Sekunden wählen.

Während der Messung gibt "sar" standardmäßig die CPU-Benutzung aus:

10:02:17	%usr	%sys	%wio	%idle
10:02:22	3	34	11	52
10:02:27	2	20	12	66
10:02:32	1	29	70	0
10:02:37	4	30	49	17
	...			
10:03:07	4	25	15	56
Average	2	24	22	51 Mittelwert

- %usr: % für CPU-Zeit im **"user-mode"**
- %sys: % CPU-Zeit im **"system-mode"**
- %wio: warten auf **"block"**-E/A
- %idle: CPU war untätig

Auswertung der Messdaten:

```
sar -r -f mess_dat            #Beispiel: freier Speicher & "swap"
```

10:02:17	freemem	freeswp
10:02:22	410	7691
10:02:27	472	7757
10:02:32	280	7756
10:02:37	365	7715
...		
10:03:07	476	7732

jeweils in 4 KB Seiten !

Auswertung der Messdateien:

sar -d -f mess_dat # Beispiel: Platten-Aktivität

10:02:17	device	%busy	avque	r+w/s	blks/s	avwait	avserv
10:02:22	dsk-0	1	1.0	0	4	0.0	25.0
	dsk-1	1057	3.9	45	359	677.1	233.6
10:02:27	dsk-0	3	1.0	1	11	0.0	21.4
	dsk-1	1068	12.1	46	362	2594.4	234.6
			...				
			...				
10:03:02	dsk-1	1287	6.6	50	402	1424.1	255.8
10:03:07	dsk-0	3	1.0	1	11	0.0	18.6
	dsk-1	1128	8.6	48	377	1807.9	236.4
Average	dsk-0	3	1.0	2	12	0.1	21.9
	dsk-1	1163	8.1	48	384	1705.4	240.2

Datenüber-
tragungszeit

mittlere Anzahl von
Transfers, die
warten mußten

Anzahl
Transfers
pro Sekunde

Anzahl
Blöcke

mittlere
Zeit in ms,
die ein
Transfer

warten
mußte

gedauert
hat

truss: Systemaufrufe protokollieren

Aufruf:

truss -o protdat KOMMANDO # Kommando starten und
protokollieren

oder

truss [-o protdat] -p PID # laufenden Prozeß mit
NR.: PID protokollieren

Beispiele:

a) ***truss -o protdat ced testdat***
pg protdat # Protokolldatei ansehen

b) ***truss -p 1*** *# init-Prozeß protokollieren*

Das "truss" Kommando ist beschrieben im Handbuch "Kommandos V5.4, Band 2, L-Z".

Die Auswertung der Protokolldateien erfordert allerdings sehr gute Kentnisse der C-Systemaufrufe und führt daher über das Thema dieses Kurses hinaus.

Systemgrenzen, Systemparameter

Informationsdatei "mtune"

UNIX System V.4 besitzt ca. 170 Systemparameter, die in der Datei **"/etc/conf.d/cf.d/mtune"** aufgelistet sind.

Sie finden dort die Namen der Systemparameter, die Standartwerte, Minimal- und Maximalwerte und Kommentare.

In der Datei "mtune" sollten Sie keine Änderungen vornehmen!

Konfigurationsdatei "stune"

Wollen Sie einen Systemparameter ändern, so rufen Sie das Kommando auf:

***/etc/conf/bin/idtune** NAME Wert*

"idtuen" überprüft den neuen Wert und trägt ihn in die Datei **"/etc/conf/cf.d/stune"** ein. In dieser Datei finden Sie alle Systemparameter, deren Wert vom Standardwert abweichen soll.

Habne Sie einen Wert geändert, müssen Sie den Systemkern neu binden. Dazu rufen Sie ein Kommando auf:

```
/etc/conf/bin/idbuild
init 6                  # System neu starten
```

Einige Systemgrenzen entfallen:

- Einige typische Systemgrenzen früherer UNIX-Systeme entfallen in SINIX V 5.4, da die betreffenden Systemtabellen jetzt dynamisch angelegt werden.

- Andere Systemgrenzen könne Sie in V5.4 prozeßspezifisch über den C-System-aufruf **"setrlimit ( )"** setzen und mit **"getrlimit ( )"** abfragen.

- Eine genaue Beschreibung der Systemparameter finden Sie im Handbuch: "Leitfaden für Systemverwalter".

Besonderheiten des MX500

Bedienung des Hardware-Monitors

Zeichensatz und Uhrzeit einstellen

Der Hardwaremonitor meldet sich mit einem "*" an der ausgewählten Konsole (entweder C oder S).

kd0 deutschen Zeichensatz für lokale Konsole einstellen

kd1 deutschen Zeichensatz für Service-Konsole einstellen

ki0 internationalen Zeichensatz für lokale Konsole (Standardeinstellung)

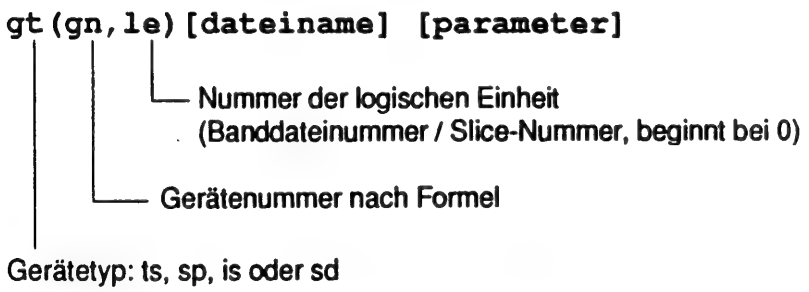
wd=jjmmddhhmm.ss Hardware-Uhr stellen (Greenwich Maintime)

rd Hardware-Uhr lesen ("read date")

ra Bootflags lesen

Gerätenamen und Nummern im Monitor

Der Hardwaremonitor spricht Geräte nicht über ihre SINIX-Gerätedateinamen an, sondern verwendet Namen in einer betriebsystemunabhängigen- **(bsu-)-Darstellung:**



ts — Magnetband-Kassetten-Laufwerk am SCSI-Bus
sp— neue ("ssd" bzw. SPA)-Platten am synchronen SCSI-Bus (MX500-90)
is — ESDI-Platten am Multibus (MX500-75)

Typischer Bootstring für MX500-90:

b 0 sp(0,10)unix root=ssd(1) swap=ssd(2) rootfstype=ufs

Besonderheiten des MX500

Gerätenamen und Nummern

Formel für Platten am synchronen SCSI-Bus (MX500-90):

gn = Kanal * 64 + Anschluß * 8

Formel für Platten am Multibus (MX500-75):

gn = Multibus * 512 + Controller * 8 + Anschluß

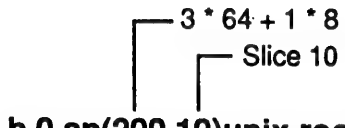
bsu-Darstellung	SINIX-Notation	Bedeutung
ts(32,1) Streamer hat immer Nr. 32	-	Banddatei Nr. 1 (bei 0 beginnend) des Streamerbandes
MX500-90:		
sp(0,10)	c0d0s10	/stand-Bereich der 1. Platte
sp(8,10)	c0d1s10	Slice 10 der 2. Platte
sp(64,1)	c1d0s1	Slice 1 der 1. Platte am 2. Kanal
MX500-75:		
is(0,0)	is0a	a-Bereich der 1. "is"-Platte Platten-Controller Nr.: 0
is(1,0)	is1a	a-Bereich der 2. "is"-Platte am 1. Controller (Anschluß: 1)
is(8,0) | 0 * 512 + 1 * 8	is2a	a-Bereich der 2. "is"-Platte mit eigenem Controller Multibus-Adapternummer 0, Platten-Controller Nr.: 1 Anschluß am Controller: 0
is(16,0) | 0 * 512 + 2 * 8	is4a	a-Bereich einer "is"-Platte mit Multibus-Adapternummer 0, Platten-Controller Nr.: 2 Anschluß am Controller: 0

Besonderheiten des MX500

bsu-Programme booten

MX500-90:

Booten von einer Platte am 3. Kanal, 2. Anschluß:



b 0 sp(200,10)unix root=ssd(3201) swap=ssd(3202) rootfstype=ufs

ssd(Minornummer)

= Kanal * 1024 + Anschluß * 128 + Slice

Booten vom Installationsband:

b 0 ts(32,1) root=ramd(0) swap=ramd(1)

Banddatei Nr. 1

Besonderheiten des MX500

MX500 - Die wichtigsten Daten im Vergleich

MX500 Modell 75

Prozessoren (NSC)
- Dualprozessor-Boards
- NSC 32532, 25MHz
  Cache je Prozessor:
  On-chip 1,5 Kbyte,
  On-board 64 Kbyte
- Bis zu 8 Prozessoren
- Prozessorleistung 16 - 64 Mips

Hauptspeicher (1 Mbit)
- 16 bis 64 Mbyte

5 1/4"-Festplatten
- Anschluß über ESDI-Controller an Multibus I
- 380/760 Mbyte Laufwerke
- Bis zu 6 ESDI-Controller
- Max. 12 Laufwerke
  (6 im GS, 6 im ES)
- Bis zu 9,1 Gbyte

5 1/4" Floppy Disk

SINIX V5.2x
- UNIX V.2 für Multiprozessorsystem

MX 500 Modell 90

Prozessoren (Intel)
- Monoprozessor-Boards
- Intel 80486, 25 MHz
  Cache je Prozessor:
  On-chip 8 Kbyte,
  On-board 128 Kbyte
- Bis zu 7 Prozessoren
- Prozessorleistung 30 - 105 Mips

Hauptspeicher (4 Mbit)
- 32 / 64 / 128 Mbyte

5 1/4"-Festplatten
- Anschluß über SCSI-Systembus-Adapter an schnellen Systembus
- 760 Mbyte/1,5 Gbyte Laufwerke
- 1 oder 2 SCSI-Systembus-Adapter jeweils mit 2 oder mit 4 Kanälen
- Bis zu 24 Laufwerke
  (12 im GS, 12 im ES)
- Bis zu 36 Gbyte

3 1/2" Floppy Disk

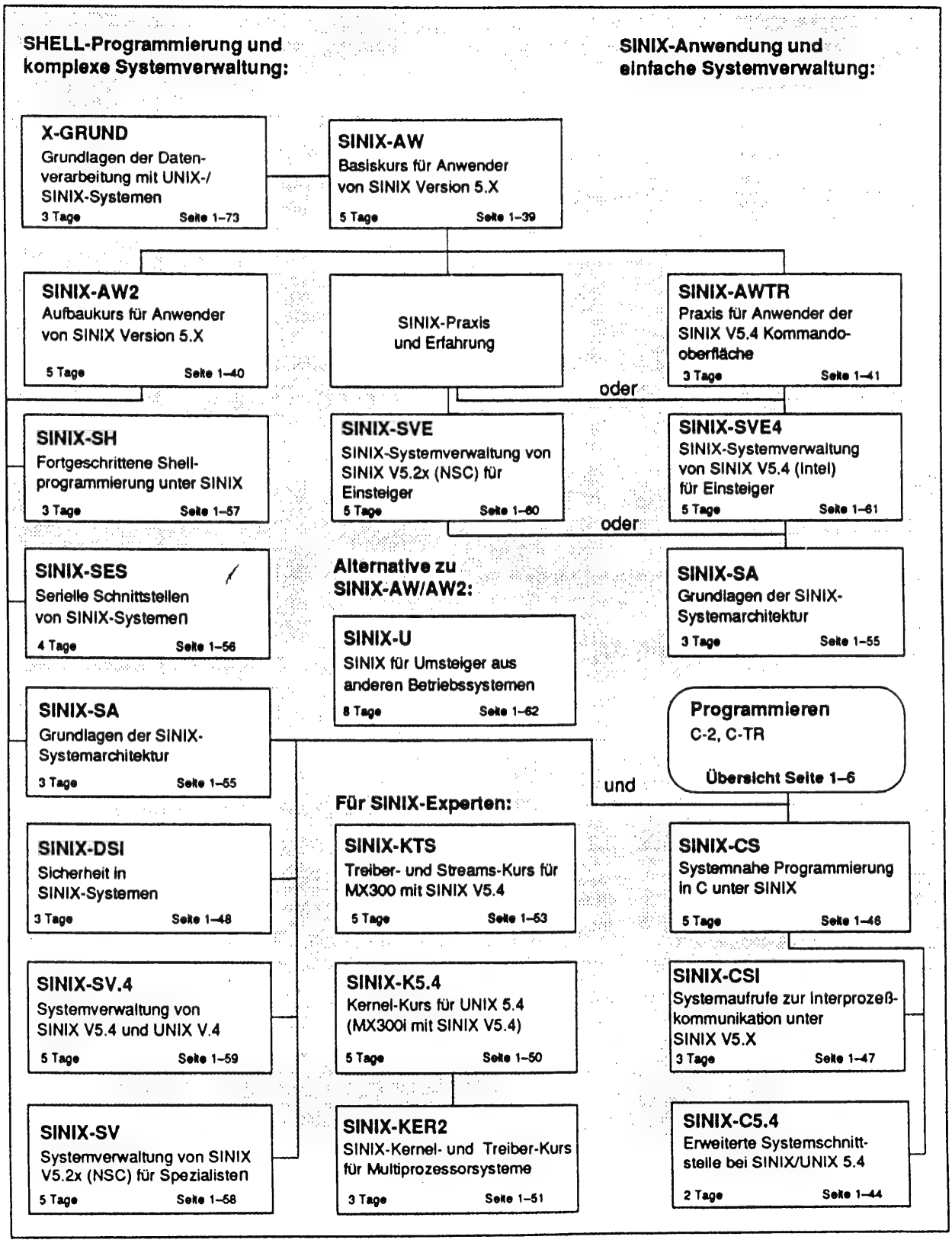
SINIX V5.40
- UNIX V.4 für Multiprozessorsystem

Umrüstmöglichkeit für MX500 Modell 75 auf Intel-Prozessoren

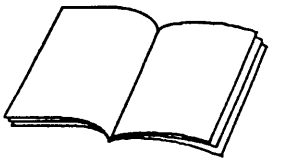
Auch Besitzer eines MX500-75 können die Intel 80486 Prozessoren nutzen. Der MX500-75 kann nachträglich mit Intel-Prozessoren und SINIX V5.40 ausgestattet werden. Damit steht auch diesen Anwendern die Welt von UNIX System V Rel. 4 offen. Eine komplette Hochrüstung auf alle Eckwerte des MX500-90 ist allerdingst nicht möglich. Anstelle der bisher möglichen 4 Dualprozessorboards des MX500-75 können bis zu 4 Singleprozessorboards mit Intel 80486 eingesetzt werden. Alle anderen Maximalwerte des MX500-75 (HSP-Ausbau/ Festplatten-Kapazität / max. Schnittstellen) bleiben gleich.

SINIX-Betriebssystem

Sie wollen auf der Betriebssystem-Ebene arbeiten?
Wir bieten Ihnen dazu Kurse für Einsteiger, Fortgeschrittene und Spezialisten, die Konzepte für Systemsicherheit entwerfen, die SINIX-Systemschnittstelle nutzen und selbständig mit dem Kernel arbeiten.



Anhang



Literaturhinweise

- Pflichtlektüre für den SINIX-Systemverwalter sind die SINIX-Freigabemitteilung und die Manuale der Angebotseinheiten:

"Messagefile" : SINIX-SPR-D (für deutsch)

Freigabemitteilung gedruckt und auf Diskette mit
Manualkorrekturen
Bediensystem für Systemverwalter

Dokumentationspaket: SINIX-DOC-D (für deutsch)

Kommandos Teil 1 bis 3
COLLAGE-Bediensystem
SINIX SPOOL V3.0 / V3.1
Referenzhandbuch für den Systemverwalter
Leitfaden für den Systemverwalter

Das Handbuch "Leitfaden für den Systemverwalter" beschreibt UNIX System V.4 . SINIX V5.4 weicht jedoch in einigen Punkten von dieser Dokumentation ab (z.B. in der Terminalverwaltung).

Die beiden Angebotseinheiten müssen Sie gesondert bestellen

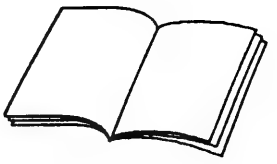
Zum Umstieg von SINIX V5.2x auf SINIX V5.4 kann Ihnen der "Portierungs-leitfaden" (Band 1) gute Dienste leisten.
Bestellnummer: U 6313-J-Z99-1

Weitere Literatur für Systemverwalter basiert meist auf UNIX System V.3 oder auf Berkeley- bzw. Sun-Systemen.Sie benötigen daher einige Kenntnisse, um die Informationen herauszufiltern, die auch für SINIX V5.4 gültig sind.

Beispiele:

- UNIX Systemverwaltung/ Hauke Richter / Addison-Wesley (Deutschland) 1991/ ISBN 3-89319-297-2 (Beschreibung von UNIX System V.3)

- UNIX System Administration Handbook/ E. Nemeth, G. Snyder, Scott Seebass/ Prentice-Hall 1989/ ISBN 0-13-933441-6



Nachschlagewerke für den Systemverwalter:

- SINIX V5.4 Kommandos: A - K und L - Z
  für Benutzer und Systemverwalter

- Referenzhandbuch für Systemverwalter
  ein Handbuch mit mehreren Kapiteln:

  - Kommandos für Systemverwalter z.B. fsck (1M)

    (Kapitel 1 der klassischen UNIX-Gliederung)

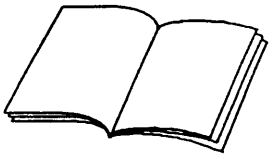
  - Dateiformate z.B. passwd (4)

    (Kapitel 4 der UNIX-Gliederung)

  - Include Dateien z.B. vtoc.h (5)

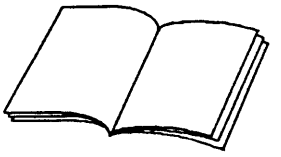
  - Geräte z.B. fd (7)

Vorsicht: In der ersten Ausgabe dieser deutschen Handbücher haben sich einige Übersetzungsfehler eingeschlichen.



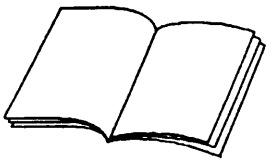
cleanup

```
#! /sbin/sh
#  private cleanup-Prozedur fuer SINIX V5.4
#  1 mal in der Woche aufrufen: Freitag 12:00
#            z.B.: mit "crontab"-Kommando !!!
#
#  am Freitag Nachmittag sollte der Rechner
#  dann heruntergefahren werden
#
# PATH darf nicht mit Doppelpunkt beginnen !!!!
PATH=/sbin:/usr/bin:/usr/sbin:/opt/bin:/opt/etc:/usr/ucb
export PATH
#
# Aufraeumarbeiten unter /var/adm
cd /var/adm
#
cp klog.msg Oklog.msg      # letzte Version aufbewahren
> klog.msg                 # Datei auf 0 setzen
#
mv sulog Osulog
> sulog                    # Datei auf 0 setzen
#
cp wtmp Owtmp              # letzte Version
cp wtmpx Owtmpx            # letzte Version
> wtmp
> wtmpx
#
# Aufraeumarbeiten: /var/cron/log und /var/crash
#
cp /var/cron/log /var/cron/Olog
> /var/cron/log
#
rm -f /var/crash/*
#
# Aufraeumarbeiten: /tmp und /var/tmp
#
cd /tmp
find . -type f -mtime +3 -exec rm {} \;
cd /var/tmp
find . -type f -mtime +3 -exec rm {} \;
#
# alte core-Dateien im gesamten System loeschen
# (find-Aufruf dauert sehr lange)
cd /
find / -name core -atime +7 -exec rm {} \;
```

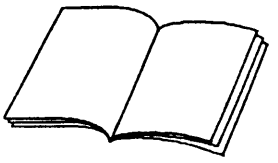


save_vtoc Platteneinteilung sichern

```
#! /bin/sh
# save_vtoc: V1.0 13.11.91
#######################################################################
# Prozedur zur Sicherung der Platteneinteilungen in Dateien,
# die als Eingabe fuer "disksetup -d" verwendbar sind.
#
# Siemens Nixdorf Training Center
#      D9 DS TC 21
#
# Ausgabedateien fuer jede Platte:
#
#      VTOC.c?d? Ausgabedateien von prtvtoc -f
#      DISK.c?d? (verwendbar fuer disksetup -d)
#      DISK.0, DISK.1 (Verweise auf DISK.c?d?)
#######################################################################
# temporaere Dateien loeschen
trap '/bin/rm [a-z]*.c[01]d[0-1]' 0 2 3
#
# Feststellen, welche Platten vorhanden sind, VTOC.c?d? erzeugen:
cd /tmp
DISKS=
for d in c0d0 c0d1 c1d0 c1d1
do
     if prtvtoc -f VTOC.$d /dev/rdsk/${d}s0 2>/dev/null
     then
       DISKS="$DISKS $d"
       # Tabulatoren entfernen
       tr -s '\11' ' ' < VTOC.$d > vtoc.$d
     else
       rm VTOC.$d
     fi
done
#
# fuer jede Platte Standardwertdatei DISK.c?d? erstellen
#
i=0
for d in $DISKS
do
   # Ueberschrift, Slice 0, Slices mit Groesse 0 und
   # Slice 7 aus vtoc.c?d? entfernen:
ed - vtoc.$d <<\stop
1,2d
1,$s/^ //
g/ 0$/d
/7 /d
w
q
stop
   #
```

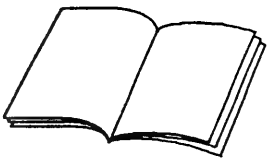


```
  # In /etc/vfstab nachsehen, welcher slice wo montiert ist
  # unterschiedliche Behandlung wegen root-slice auf 1. Platte
  case $d in
  c0d0) tr -s '\11' ' ' < /etc/vfstab |
          sed -n "/rroot/{s:/dev/root:1:
                          s: / : /root :p
                          }
                  /$d/s:/dev/dsk/${d}s::p" > vfstab.$d ;;
  *) sed -n "/$d/s:/dev/dsk/${d}s::p" /etc/vfstab > vfstab.$d ;;
  esac
  #
  # std_datei zusammenbauen und Zeilen aus vtoc-Datei loeschen
  > std_datei.$d
  while read SLICE RAW DVZ DSTYP REST
  do
    # Slice-groesse aus vtoc.$d heraussuchen
    set `grep "^$SLICE " vtoc.$d` 0 0 0 0 0
    echo "$SLICE $DVZ $DSTYP $5" >> std_datei.$d
    # Zeile in vtoc.$d loeschen
    grep -v "^$SLICE " vtoc.$d > tmp.$d
    mv tmp.$d vtoc.$d
  done < vfstab.$d
  #
  # Alle Bereiche hinzufuegen, die nicht in der /etc/vfstab
  # sondern nur in der vtoc-Datei stehen
  while read SLICE TAG FLAGS START SIZE
  do
    case $TAG in
    *3) echo "$SLICE swap - $SIZE" >> std_datei.$d ;;
    *) echo "$SLICE r$SLICE - $SIZE" >> std_datei.$d ;;
    esac
  done < vtoc.$d
  #
  # Auf jeder Platte einen Slice variabel mit "100 W" angeben,
  # um kleine Unterschiede in den Plattengroessen auszugleichen
  #
  egrep -v '/root|/usr|/stand|/opt|/tmp|swap' std_datei.$d > nonsystem.$d
  #
  ZAHL=`cat nonsystem.$d | wc -l`
  if [ "$ZAHL" -gt 0 ]
  then MAX=`awk '$4 > max { name=$2; max=$4 }
                    END { print name }' nonsystem.$d`
  else MAX=/xxxx
  fi
  # megabytes ausrechnen
  awk '$2 != MAX { MB = $4 / 2048
       printf"%s\t%s\t%s\t%3.0fM\n",$1,$2,$3,MB
       }
       $2 == MAX {
       printf"%s\t%s\t%s\t100W\n",$1,$2,$3
       }' MAX=$MAX std_datei.$d | sort +0n -1 | > DISK.$d
  ln DISK.$d DISK.$i
  i=`expr $i + 1`
done
```

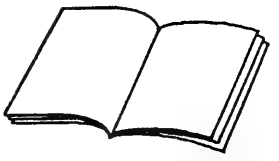


system_sich Systemsicherung

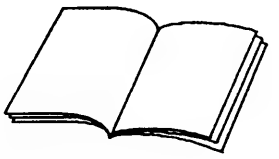
```
#! /sbin/sh
# system_sich V2.0 18.11.91
#############################################################
# Prozedur zur Systemsicherung in Verbindung mit "save_vtoc"
#                  und einer eigenen Prozedur "cleanup"
#
# sichert:    Platteneinteilung (siehe "save_vtoc")
#             und Label                                0. Banddatei
#             "/" und "/stand"                         1. Banddatei
#             "/usr" und "/var"                        2. Banddatei
#             "/opt"                                   3. Banddatei
#             /home/oasys,/home/vmsys,/home/tele       4. Banddatei
#
# Der Rest von /home muss getrennt gesichert werden (z.B. mit DSX)
# Falls "/opt" zu gross ist, muss anstelle der 3. und 4. Banddatei
# ein neues Band verwendet werden !!!
#############################################################
PATH=/sbin:/usr/bin:/usr/sbin:/usr/ucb:/opt/bin
export PATH
exec > /dev/sysmsg 2>&1      # Ausgabe auf Konsole umlenken
# Ist Band eingelegt ?
sync
mt -f /dev/tape noret || { echo "Kein Band eingelegt"; exit 1 ; }
mt -f /dev/tape ret & # Straffen im Hintergrund
#
cd /
sync
init 1              # in run level 1 herunterfahren
sleep 70
killall 15          # alle Prozesse sanft beenden
sleep 5
killall             # restliche Prozesse brutal beenden
sleep 5
#
wait                # auf mt-Kommando warten
#
echo "Dateisysteme abhaengen"
umountall           # fast alles abmontieren
umount /proc        # wichtig !!
umount /dev/fd >/dev/null 2>&1   # eventuell unnoetig
#
# Dateisysteme wieder montieren:
mount /stand >/dev/null 2>&1     # nur zur Sicherheit
mount /usr     # "find" und "cpio" stehen im /usr-Bereich
mount /var
mount /opt
mount /tmp     # fuer Protokoll der Sicherung
#
```



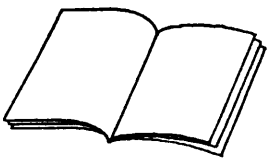
```
#
exec > /tmp/Protokoll 2>&1   # Protokolldatei
cat /dev/klog &              # Kernelmeldungen mitlesen
PID=$!
tail -f /tmp/Protokoll >/dev/sysmsg 2>&1 & # Protokoll auf Konsole ausgeben
PID="$PID $!"
#
echo "Sicherungsprotokoll `date`"
#
# aufraeumen:
echo "/var/crash, /var/adm, /var/cron und /tmp bereinigen"
cleanup
#
# Platzbedarf berechnen, Platteneinteilung feststellen,
# Label erzeugen, alles in 0. Banddatei sichern
#
cd /tmp
echo    "Systemsicherung vom `date` im cpio -oc Format
        Platteneinteilung und Label    0. Banddatei
        "/" und "/stand"               1. Banddatei
        "/usr" und "/var"              2. Banddatei" > Label
#
# Platzbedarf berechnen
#
MB1='df -k | egrep "root|stand|usr|var|opt" |
          awk '{ summe = summe + $3 }
               END { printf"%3.0f", summe/1024 }''
MB1='echo $MB1'
MB2='df -k | egrep "root|stand|usr|var" |
          awk '{ summe = summe + $3 }
               END { printf"%3.0f", summe/1024 }''
MB2='echo $MB2'
if [ "$MB1" -gt 135 ]
then
  echo "Zu viele Daten: $MB1 (einschl. /opt)"
  echo "/opt passt nicht auf das Band"
  echo "Sicherungsumfang: $MB2 MB"
  OPT=N
else
  echo "      /opt          3. Banddatei" >> Label
  echo "Sicherungsumfang: $MB1 MB"
  OPT=J
fi
if [ "$MB2" -gt 135 ]
then
  echo "Zu viele Daten: $MB2 (ohne /opt)"
  df -k
  echo "**** Abbruch *******"
  exit 1
fi
```



```
if [ "$MB1" -lt 125 ]
then
   echo " /home/oasys,vmsys,tele 4. Banddatei" >> Label
   HOMESYS=J
else
   HOMESYS=N
fi
cat Label
#
echo "Platteneinteilung sichern:"
rm -f DISK.* VTOC.*
#
save_vtoc || exit 1
#
ls Label DISK.* VTOC.* | cpio -ocBO /dev/tapen
#
# /, /stand in 1. Banddatei sichern:
#
echo "Sicherung von / und /stand :"
#
# Liste aller Dateien im / Dateisystem ohne /usr/*,
#                                    /var/*, /tmp/* erzeugen:
cd /
for d in 'ls -a'
do
   case $d in
   .) echo . ;;     # wegen der Zugriffsrechte
   ..) : ;;
   usr|var|opt|tmp) echo $d ;; # nur das leere Verzeichnis
   home*) echo $d ;;
   *) if test -d $d          # bei Verzeichnissen
      then find $d -print    # "find" aufrufen
      else echo $d           # nur Dateinamen ausgeben
      fi ;;
   esac
done | cpio -ocBO /dev/tapen
#
# /usr und /var sichern:
echo "Sicherung von /usr und /var:"
find usr var -print |cpio -ocBO /dev/tapen
#
#
if [ "$OPT" = "J" ]
then
#    /opt sichern ********************************
   echo "/opt : "
   find opt -print |cpio -ocBO /dev/tapen
fi
```



```
if [ "$HOMESYS" = "J" ]
then
#    /home/vmsys,oasys,tele sichern **************************
  echo "/home/vmsys,oasys,tele:"
  mount /home
  find home/vmsys home/oasys home/tele -print | cpio -ocBO /dev/tapen
fi
# zurueckspulen
< /dev/tape
echo "Ende der Sicherung `date`"
sleep 2
kill $PID
mail root < /tmp/Protokoll
cp /tmp/Protokoll /Protokoll.$$
sync
sleep 10
init 6   # reboot
```



Spiegelplatten (VPSS)

Lit.: SINIX V5.40/Virtuelles Platten-Subsystem
Benutzerhandbuch, U6768-J-Z145-1

Ziel:

Bei einem Fehler oder Ausfall der Hardware (z.B. Plattenfehler) sollen:

- keine Datenverluste auftreten
- das Computer-System sollte in möglichst kurzer Zeit wieder einsetzbar sein

Lösungsansatz:

Alle Daten müssen auf mehreren Platten mit eigenen
Platten-Controllern gleichzeitig abgespeichert werden.

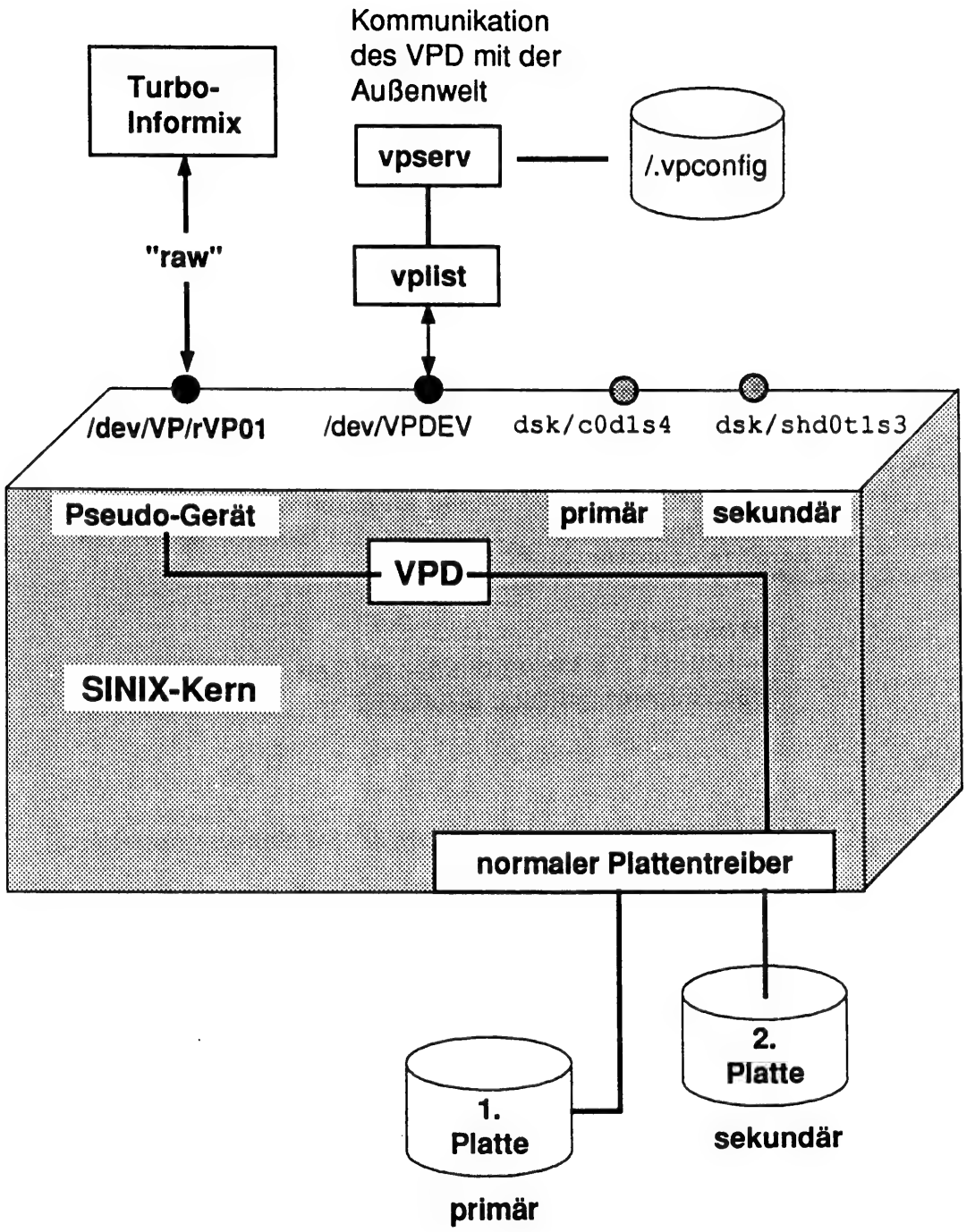
- Bei System- und Anwendersoftware und bei Konfigurations-Dateien, die sich selten ändern (z.B. "root"- und "usr"-Bereich), können Sie die betreffenden Dateien oder Plattenbereiche nach jeder Änderung auf eine Sicherungsplatte kopieren. Zu diesem Zweck benötigen Sie kein Spiegelplatten-System.

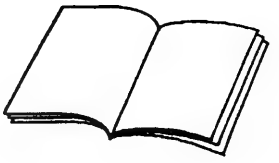
- Bei Datenbanken, die laufend verändert werden, sollte **jede Veränderung gleichzeitig auf mehreren Platten erfolgen.**

  Das **"Virtuelle Platten-Subsystem"** (VPSS) ermöglicht diese Datenvervielfachung für Anwendungen, die **direkt mit "raw"-Gerätedateien** von Plattenbereichen arbeiten (z.B. Turbo-Informix, Informix-ONLINE oder Oracle).
  Für Plattenbereiche, die eine Dateisystem enthalten, ist VPSS erst ab SINIX V5.41 einsetzbar !

  Anstatt mit einer normalen "raw"-Gerätedatei müssen Sie die Anwendung mit einem **Pseudogerät** (virtuelle Platte) verbinden. Der VPD (VP-Treiber) im SINIX-Kern leitet alle Daten, die das Pseudogerät erhält, gleichzeitig an einen primären und eventuell mehrere sekundäre Plattenbereiche (Kopien) weiter.

  Die Zuordnung zwischen Pseudogeräten, primären und sekundären Plattenbereichen muß in einer binären Datei "/.vpconfig" hinterlegt sein.


Kommunikation
des VPD mit der
Außenwelt
Turbo-
Informix
vpserv
/.vpconfig
"raw"
vplist
/dev/VP/rVP01
/dev/VPDEV
dsk/c0d1s4
dsk/shd0t1s3
Pseudo-Gerät
primär
sekundär
VPD
SINIX-Kern
normaler Plattentreiber
1.
Platte
primär
2.
Platte
sekundär



Dateiverzeichnisse und Dateien für VPSS:

`/sbin/VP/..`

Dateiverzeichnis mit allen Kommandos und Dienstprogrammen des VPSS

`/dev/VP/..`

Dateiverzeichnis für Pseudo-Geräte:

rVP01 erste Pseudogerätedatei
rVPmax letzte Pseudogerätedatei

Weitere Pseudo-Gerätedateien müssen Sie mit dem "mknod"-Kommando anlegen. Sie müssen den Typ "c" und die gleiche Majornummer angeben wie bei "rVP01". Die Minornummer muß zwischen den Minornummern von "rVP01" und "rVPmax" liegen.

```
cd /dev/VP
ls -l rVP01    # Majornummer merken
/sbin/mknod rVP02 c MAJOR 2
```

`/dev/VPDEV`

Gerätedatei für "vpserv" und "vplist" zur Kommunikation mit dem VP-Treiber (VPD).

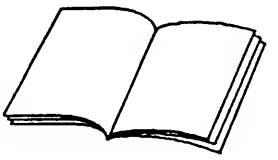
`/etc/inet/services`

Das VPSS bedient sich der gleichen Mittel zur Interprozeß-kommunikation wie LAN1 (TCP/IP). Für VPSS müssen zwei zusätzliche Zeilen in der Datei "/etc/inet/services" Ihres Netzverwaltungsrechners vorhanden sein:

```
vpserv_cmd 1230/tcp
vpserv_brd 1232/udp
```

Fehlen diese beiden Zeilen, so müssen Sie sie ergänzen und anschließend folgende Kommandos eingeben:

```
cd /var/yp
make services
```



Beispiel für eine ASCII-Konfigurationsdatei

Im folgenden Beispiel wird 1 Plattenbereich gespiegelt:

- Pseudogerät: /dev/VP/rVP01
  Primärer Bereich: /dev/dsk/c0d1s4
  Sekundärer Bereich: /dev/dsk/shd0t1s3 (Kopie)

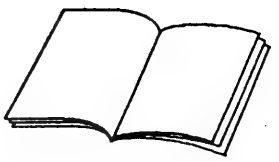
Achtung: Sie müssen hier die "block"-Gerätedateien eintragen, obwohl VPSS auf die "raw"-Gerätedateien zugreift.

Tritt ein Schreibfehler auf, deaktiviert VPSS den fehlerhaften Bereich und schaltet ggf. auf einen anderen primären Bereich um.

Beispiel: ASCII-Datei VPCONFIG

```
# Pseudo-Geraet unter /dev/VP/..
rVP01 SHADOW
        /dev/dsk/c0d1s4 /dev/dsk/shd0t1s3
        PLEN=228960
#
```

Mit PLEN geben Sie die Länge des Plattenbereichs in Blöcken von 512 Bytes an.



VPSS starten/konfigurieren:

Im Verzeichnis "/sbin/VP" steht eine "rc"-Prozedur zur Verfügung, die als Beispiel dienen kann.

Erweitern Sie zunächst Ihre PATH-Variable:

```
PATH=/sbin/VP:$PATH
export PATH
```

Falls Sie eine Ereignisprotokollierung wünschen, können Sie zunächst das Programm "vpmsg" starten:

```
vpmsg /dev/console &            # Beispiel !
```

Anschließend starten Sie die Hintergrundprozesse, die Ihnen eine Kommunikation mit dem VP-Treiber ermöglichen:

```
vpserv -b /home1/.vpconfig &        # Beispiel !
```

Der "vpserv"-Prozeß startet automatisch den "vplist"-Prozeß. Die Datei "/home1/.vpconfig" dient in diesem Beispiel als Duplikat der binären Konfigurationsdatei "/.vpconfig".

Falls Sie noch nicht konfiguriert haben, erhalten Sie die Fehlermeldung, daß die Datei "/.vpconfig" fehlt. Die Fehlermeldung ist beim ersten Starten normal. Sind die Prozesse "vpserv" und "vplist" aktiv, so können Sie Ihre ASCII-Konfigurationsdatei in das binäre Format umsetzen und gleichzeitig in den Kern laden:

```
vpreq -c VPCONFIG # ASCII-Datei: VPCONFIG
```

Sie können sich den aktuellen Status des VPD ansehen mit dem Kommando:

```
vpreq -q
```

Achtung: "vpreq" funktioniert nur, wenn "vpserv" und "vplist" aktiv sind !

Virtuelle Partition nach Fehler wiederherstellen:

```
vpreq -b rVP01
```

Benutzer und Benutzergruppen

Benutzerkennungen:

Der Systemverwalter sollte für jede Person, die am Rechner arbeiten darf, eine eigene Benutzerkennung einrichten.

Prinzipiell könnten mehrere Personen gleichzeitig unter der gleichen Benutzerkennung arbeiten. Erfahrungsgemäß treten dabei jedoch meist Probleme oder Fehler auf.

⇨ pro Person eine eigene Benutzerkennung !!!

Welche Informationen speichert SINIX für jede Benutzerkennung ?

- Benutzername	nur zum Anmelden und zur Kosmetik
- Kennwort	zum Anmelden (login)
- Benutzernummer	SINIX arbeitet intern nur mit der Benutzernummer nicht mit dem Namen !!
- Gruppennummer	—> siehe Benutzergruppen
- Login-Dateiverzeichnis	
- Startprogramm/Prozedur Login-Shell	

Benutzergruppen:

Beim Einrichten einer Benutzerkennung müssen Sie angeben, zu welcher Benutzergruppe der neue Benutzer gehören soll.

Bevor Sie eine Benutzerkennung einrichten, sollten Sie daher mindestens eine Benutzergruppe einrichten !!

Die Zugehörigkeit zu einer Benutzergruppe ist wichtig für die Zugriffsrechte auf Dateiverzeichnisse, Dateien und Programme.

Mit den Kommandos "**chmod**" und "**umask**" können Sie dafür sorgen, daß alle Benutzer einer Benutzergruppe auf gemeinsame Dateiverzeichnisse, Dateien und Programme zugreifen können.

Aufgabe 1:

a) Sie wollen als Systemverwalter das Kennwort eines Benutzers ändern. Welche Möglichkeiten haben Sie ?

Über "admin"-Bediensystem: "Benutzerverwaltung"

Mit dem "passwd"-Kommando: passwd Benutzername
(funktioniert immer !)

b) Sie übernehmen die Systemverwaltung eines vom Werk installierten SINIX-Rechners. Bei welchen Benutzerkennungen sollten Sie sofort die Kennwörter ändern ?

Bei allen Kennungen, die die Benutzernummer 0 haben
(siehe /etc/passwd): root, admin, sysadm, ...

e) Ändern Sie die Zugriffsrechte aller Dateien und Datei-Verzeichnisse unter halb Ihres Login-Verzeichnisses:

Erweitern Sie zunächst die Zugriffsrechte für Ihre Gruppe und zwar:

rw — für Dateien
rwx — für Dateiverzeichnisse

Überprüfen Sie das Ergebnis mit "ls -lR".

```
cd
chmod -R g+rw .                          # Rechte geben
chmod g+x 'find . -type d -print'        # Rechte geben
ls -lR |  pg
```

f) Sorgen Sie dafür, daß Ihre Gruppenmitglieder alle von ihnen neu angelegten Dateien lesen können.

Lösung zu f):

```
umask  027       in .profile aufnehmen
<END>
```

Aufgabe 2:

In SINIX V5.4 sind die meisten Kommandos unter anderen Pfadnamen erreichbar als in früheren SINIX / UNIX Versionen.

Aus Kompatibilitätsgründen wurden symbolische Links eingerichtet, so daß die Kommandos scheinbar unter den früheren Pfadnamen ansprechbar sind.

Prüfen Sie die Pfadnamen folgender Kommandos:

- wo stehen die Kommandos in SINIX V5.4 ?
- sind die Kommandos über symbolische Links auch unter anderen Pfadnamen erreichbar ?

Kommando/ Programm	tatsächlicher Pfadname:	symbolische Verweise
sh	`/sbin/sh`	`/bin/sh , /usr/bin/sh , /usr/sbin/sh`
ls	`/usr/bin/ls`	`/bin/ls`
tar	`/usr/sbin/tar`	`/bin/tar , /usr/bin/tar`
ps	`/sbin/ps`	`/bin/ps , /usr/bin/ps , /usr/sbin/ps`
passwd	`/usr/bin/passwd`	`/bin/passwd`

Lösung zur Zusatzaufgabe: Symbolische Verweise

Normaler Verweis

- Zusätzlicher Dateiname (Verweis) für dieselbe Datei (gleiche Indexnummer) der Verweiszähler wird hochgezählt
- nur innerhalb eines Dateisystems möglich und nicht auf Dateiverzeichnisse anwendbar
- Datei ist erst dann gelöscht, wenn der letzte Verweis auf den Indexeintrag gelöscht ist

Symbolischer Verweis

- spezielle zusätzliche Datei, die den Pfadnamen der Originaldatei enthält (vom Systemkern ausgewertet)
- der Pfadname der Originaldatei muß entweder absolut oder relativ zur Verweis-Datei angegeben werden:

```
cd $HOME
ln -s /etc/passwd testdat          # richtig!!!
cat testdat
```

- eigene Indexnummer, **kein** Verweiszähler
- kann beliebig über Dateisystemgrenzen hinweggehen und ist auch auf Dateiverzeichnisse anwendbar
- es gibt keinen Schutz gegen das Löschen der Originaldatei (die Originaldatei "weiß" nichts vom SL)

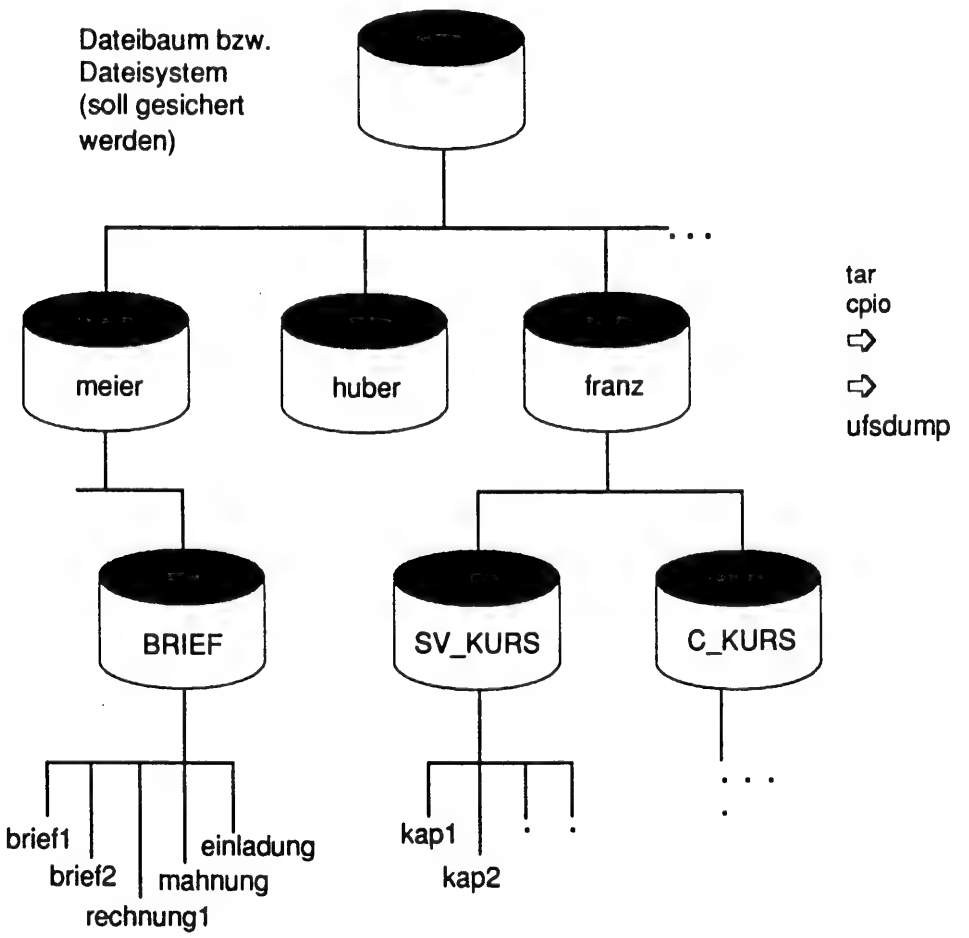
Grundprinzip von tar, cpio und ufsdump

Die Programme "**tar**", "**cpio**" und "**ufsdump**" dienen unter anderem zur Sicherung von Daten auf einem Magnetband.

tar, cpio **sichern beliebige Dateibäume**
oder einzelne Dateien

ufsdump **sichert komplette "ufs"-Dateisysteme (nicht "s5" oder "bfs")**
⇨ Einlesen mit: ufsrestore

Was machen diese Programme mit den Daten ?



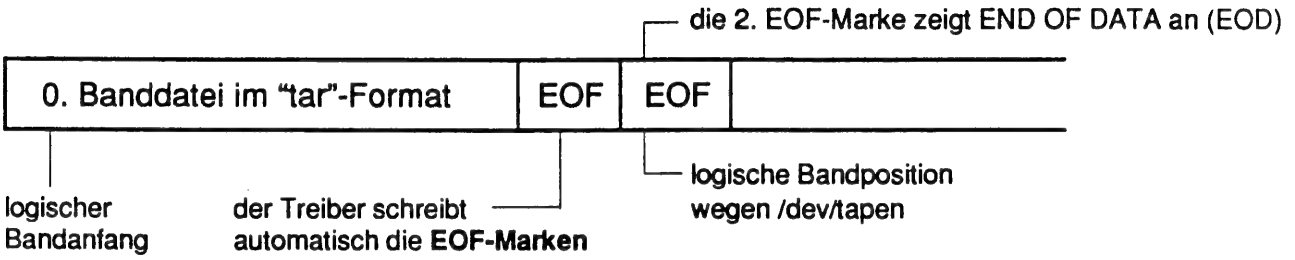
"tar", "cpio" und "ufsdump" verpacken den ganzen Dateibaum bzw. das Dateisystem in einer großen Sicherungsdatei !!

Diese Sicherungsdatei kann auf Band oder auf Diskette geschrieben werden,
es kann aber auch eine normale Plattendatei sein !

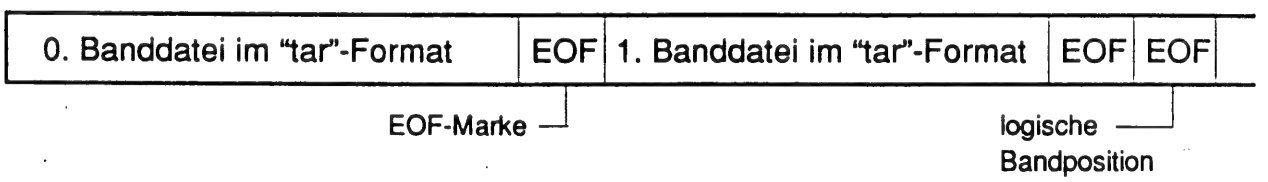
Erzeugen mehrerer Banddateien

Beispiel:

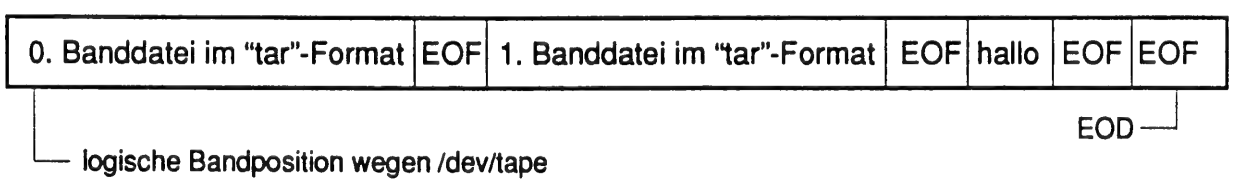
```
cd /home/meier
tar cvf /dev/tapen .          # tar erzeugt eine große Datei
                              # und schreibt sie auf das Band
```




```
cd /home1/mueller
tar cvf /dev/tapen .
```



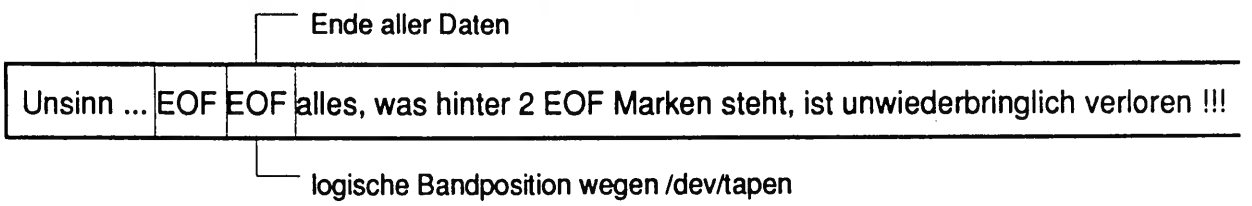

```
echo hallo > /dev/tape
```



Was passiert mit den Daten, wenn Sie jetzt eingeben :

```
echo Unsinn > /dev/tapen ? ? ?
```

⇨ In der 0. Banddatei steht "Unsinn ...", und alle
Daten sind verloren (wegen der 2 EOF-Marken !)



Dateibaum kopieren

Teile des Dateibaums so kopieren, daß die Zugriffsrechte und die Modifikationszeiten der Dateien gleich bleiben.

Beispiel:

Sie wollen den Benutzer "**meier**", der bisher im "**/home**"-Dateisystem arbeitet, in das Dateisystem "**/home1**" umsiedeln. Der Benutzer "**meier**" soll von dieser Aktion möglichst nichts merken (keine Nachteile haben).

```
mkdir /home1/meier

cd /home/meier        # in das Quell-Dateiverz. wechseln

umask 0
find . -print  |  cpio -pdmv /home1/meier
                                  |
                                  Ziel-Dateiverzeichnis

# alte Dateien löschen
rm -r /home/meier

# symbolischen Verweis einrichten, damit der
# Benutzer nichts merkt


ln -s /home1/meier /home/meier
```

Achtung

In SINIX V5.2x ist "**cpio**" nicht geeignet um den "**root**" oder den "**usr**"-Bereich zu sichern oder zu kopieren !

In SINIX V5.4 können Sie "**cpio**" für alle Zwecke verwenden!

tar

Vor- und Nachteile von "tar"

+ **einfache Bedienung:** Sie müssen keine Liste von Dateinamen über **"find"** erzeugen

Beispiel: Sie wollen **"/home/meier"** und **"/home/mueller"** sichern.
Wie lauten die Kommandos für **"tar"**, wie für **"cpio"**?

tar:
```
cd /home
tar cvf /dev/tape meier mueller
```

cpio:
```
cd /home
find meier mueller -print | cpio -ocBV > /dev/tape
```

+ **Berücksichtigung der Bandlänge** über den Schalter **"-k"**
(eventuelle Folgebandverarbeitung)

- "boot"-Disketten enthalten nicht **"tar"** sondern **"cpio"**

- Die Liste der Dateinamen, die Sie als Argument übergeben können, ist systembedingt auf ca. 5 KB begrenzt.
Dies führt zu Problemen bei Differenzsicherungen.
Warum ?

  Typische Differenzsicherung mit **"tar"** und **"find"**:

```
find . -newer sich_dat -print > list_dat
echo `date` > sich_dat
tar cvf /dev/tape `cat list_dat`
```

  Ist "list_dat" größer als 5 KB, so kann "tar" nicht starten. Fehlermeldung: **"Argumentliste zu lang"**

Deshalb: Spezielle Option "-I" bei SINIX V5.4:

```
tar cvf /dev/tape -I list_dat
```

Mit dieser Option sind auch umfangreiche Differenzsicherungen möglich.

Grundsystem sichern

Aufgabe: 5

a) Entwerfen Sie eine Prozedur, die die Dateisysteme "/", "**/stand**" und "**/usr**" mit "**cpio**" in eine große Banddatei sichert.

b) Entwerfen Sie eine Prozedur, die die Dateisysteme "/" und "**/stand**" mit "**cpio**" in eine kleine Banddatei sichert und "**/usr**" in die 2. Banddatei.

Lösungsvorschlag zu b):

```
# als "root" anmelden !!! vorher benutzer benachrichtigen !
cd /
init 1                  # in run-level 1 herunterfahren
sleep 60
killall 15              # Prozesse geordnet beenden
sleep 10
killall                 # restliche Prozesse brutal beenden
sleep 5
umountall               # alles abmontieren
umount /proc            # wichtig !!!!
umount /dev/fd          # wichtig !!!

# "cpio" und "find" stehen im "/usr"-Bereich
mount /usr              # wichtig !!!

# wir befinden uns in "/" und benötigen eine Liste der
# Dateien, die im "/" und im "/stand"-Dateisystem
# stehen:

find . -print | grep -v "/usr/." | cpio -ocBV > /dev/tapen
#
# "/usr" mit relativen Pfadnamen sichern:
#
find usr -print | cpio -ocBV > /dev/tape
mount /var              #damit Wechsel des run-levels funktioniert
init 6                  # Reboot
```

Wenn sich /home nicht abmontieren läßt, wird /home mitgesichert!
Einen weiteren Lösungsvorschlag finden Sie im Anhang des Kursordners:
Prozedur `"system-sich"`

Aufgabe 8 (Streamer):

Bitte probieren Sie Ihre Lösung erst aus, wenn Sie vom Kursleiter die Magnetbandkassette erhalten und eingelegt haben.

Schreiben Sie folgende Prozedur:

- Spulen Sie das Band zurück, vermeiden Sie Nachspannen

```
/usr/ucb/mt -f /dev/tape noret
```

- `echo "Hallo SINIX"  'date' >/dev/tapen` # 0. Banddatei

- Sichern Sie alle Dateien Ihres HOME-Dateiverzeichnisses in der nächsten Banddatei, ohne das Band zurückzuspulen. # 1. Banddatei

```
cd                                  # in HOME-DVZ wechseln
tar cvf /dev/tapen .
```

- Sichern Sie den Inhalt von "/home/sv0" mit relativen Pfadnamen in der nächsten Banddatei:

```
cd /home/sv0
tar cvf /dev/tapen .
```

- Spulen Sie das Band zurück

```
/usr/ucb/mt -f /dev/tape rewind
```

- Lesen Sie die Banddateien 0 und 2 in einem Unterdateiverzeichnis ein.

```
cd $HOME
mkdir DVZ1
cd DVZ1
cp /dev/tapen band.0                # 0. Banddatei, kein tar-Format
cat band.0                          # ASCII-Datei
/usr/ucb/mt -f /dev/tapen fsf 1     # 1 Banddatei überspringen
tar xvf /dev/tape                   # Daten einlesen
```

Standard Tastenbelegung

Tastatur 97801	Funktionstaste in der Emulation	Tastatur 97801	Funktionstaste in der Emulation
End	Ende	F11	Alt F1
Help	Ctrl F9	F12	Alt F2
Start	Ctrl F10	F13	Alt F3
Del	Ctrl F11	F14	Alt F4
Menu	Ctrl F12	F15	Alt F5
Print	Druck	F16	Alt F6
Mode	Alt M	F17	Alt F7
⤳	Ctrl →	F18	Alt F8
⤺	Ctrl ←	F19	Alt F9
⇇	Ctrl Bild ↑	F20	Alt F10
⇉	Ctrl Bild ↓	F21	Alt F11
↖	Pos 1	F22	Alt F12
↘	Ctrl Pos 1	Shift F1	Shift F1
⇧	Bild ↑	Shift F2	Shift F2
⇩	Bild ↓	Shift F3	Shift F3
↑	↑	Shift F4	Shift F4
↓	↓	Shift F5	Shift F5
←	←	Shift F6	Shift F6
→	→	Shift F7	Shift F7
Delete CHAR	Ctrl F5	Shift F8	Shift F8
Delete WORD	Ctrl F6	Shift F9	Shift F9
Delete LINE	Ctrl F7	Shift F10	Shift F10
Insert CHAR	Ctrl F1	Shift F11	Alt Shift F1
Insert WORD	Ctrl F2	Shift F12	Alt Shift F2
Insert LINE	Ctrl F3	Shift F13	Alt Shift F3
F1	F1	Shift F14	Alt Shift F4
F2	F2	Shift F15	Alt Shift F5
F3	F3	Shift F16	Alt Shift F6
F4	F4	Shift F17	Alt Shift F7
F5	F5	Shift F18	Alt Shift F8
F6	F6	Shift F19	Alt Shift F9
F7	F7	Shift F20	Alt Shift F10
F8	F8	Shift F21	Alt Shift F11
F9	F9	Shift F22	Alt Shift F12
F10	F10		

Trimodale Tastatur

Die 97801-Terminalemulation unterstützt folgende Varianten der Trimodalen Tastatur von Siemens Nixdorf, TRIMOD-210 und TRIMOD-1891. Damit die Tastatur genutzt werden kann, ist der entsprechende Treiber vor Aufruf der Emulation zu starten. Nach Beendigung sollte der Tastaturtreiber wieder gestopt werden, da dieser Speicherplatz benötigt. Mit der Batch-Datei 97801t.bat ist dieser Vorgang automatisiert.

Beispiel der 97801t.bat

```
rem Installation des Tastaturtreibers
c:\97801\tm1891
rem Aufruf der Emulation
c:\97801\97801.exe
rem Deinstallation des Druckertreibers
c:\97801\tm1891 /u
```

Um die Funktionstasten der Trimodalen Tastatur belegen zu können, ist ein spezielles Programm zu starten. Diesem Programm muß die dazugehörige Tastenbelegungsdatei und der Terminaltyp mitgegeben werden.

Programmaufruf **makekey tm1891.key 97801**

Wie die Tasten belegt werden, ist im Kapitel 6-1 nachzulesen.

Ein Zusatzprogramm ermöglicht, daß die Tastenbelegung der Trimodalen Tastatur am Bildschirm angezeigt oder auf einem Drucker ausgegeben werden kann.Dabei ist die Tastenbelegungsdatei und der Terminaltyp anzugeben.

Ausgabe am Bildschirm **listkey tm1891.key 97801 /t**

Ausgabe auf dem Drucker **listkey tm1891.key 97801 /p**

Installation des Druckertreibers

Der Drucker-Treiber multi.c (C-Source) ist auf der Installationsdiskette im DOS-Format vorhanden. Der Treiber muß auf den SINIX-Rechner kopiert, in das SINIX-Format umgewandelt und übersetzt (kompiliert) werden. Wird der Drucker-Treiber mit dem speziellen SINIX-Kommando DOSCOPY kopiert, sollte die Datei unter einem anderen Namen kopiert werden.

```
doscp a:\multi.c multi.dos
```

Die kopierte C-Source ist nun im DOS-Format auf dem SINIX-Rechner und muß in das SINIX-Format umgewandelt werden.

```
sinfilt -d -o multi.mx multi.c
```

Anschließend kann sie mit dem Kommando cc übersetzt werden.

```
cc -o multi multi.c  (RM)
```

```
cc -o multi -DATT multi.c  (MX-Intel)
```

```
cc -o multi -DUCB multi.c  (MX-NSC)
```

Nach dem Kompilieren ist der ausführbare Drucker-Treiber in das Verzeichnis /etc zu spielen.

```
mv multi /etc
```

Damit der Drucker-Treiber automatisch gestartet wird, ist ein Eintrag in der Datei /etc/inittab erforderlich, der einen *getty* startet. Ist ein Eintrag für diese Schnittstelle vorhanden, muß eine Änderung vorgenommen werden. Desweiteren muß eine Startdatei im Verzeichnis /etc/rc2.d erzeugt werden. Die Startdatei muß mit dem Buchstaben "S" beginnen und mit einer Nummer versehen werden, mit der die Reihenfolge ihrer Ausführung bestimmt wird. In der Konfiguration der Schnittstellenparameter (siehe Seite 4-2) ist der der Eintrag Multi anzuwählen.

Eintrag in der /etc/inittab.

```
nn:2:respawn:/sbin/getty  /dev/ttys/ttyxxx
```

Beispiel einer Startdatei.*S87multi.*

```
# start des Drucker-Treiber für die n.e.t.z- Terminalemulation
/etc/multi B Baudrate ttyxxx
```

Der Drucker-Treiber legt automatisch in dem Verzeichnis /dev ein Unterverzeichnis ./printer an. In diesem Verzeichnis wird die Gerätedatei **/dev/printer/<ttyxxx>** erzeugt, die im Druck-Spooler des Rechners einzutragen ist.

Der Druckerserver der Emulation kann direkt über den SINIX-Spooler angesprochen werden.
Beispieleintrag in der Datei /usr/spool/spooler/CONFIG

D099 '/opt/etc/lppost -xon -nopoll' /dev/printer/ttyxxx -hd= .....

G099 (D099) ' PC-Drucker der Emulation'

admin (D099)

Dabei ist xxx die Nummer der Schnittstelle an dem der PC angeschlossen ist.

Diese Einträge können nicht über das Administrationsprogramm (COLLAGE) erfolgen, sondern müssen mit Hilfe eines Editors eingetragen werden. Außer in der CONFIG-Datei muß der PC-Drucker auch in den Dateien **/usr/spool/spooler/Drucker** und **/usr/spool/spooler/Gruppen** eingetragen werden. In der Datei Drucker wird der neue Drucker einfach angehängt. Ein Eintrag in der Datei Gruppen ist nur erforderlich, wenn der PC-Drucker auch einer neuen Druckergruppe zugeordnet werden soll. Nach dem die Eintragungen erfolgt sind, muß eine Übersetzung dieser Dateien z.B. mit dem Befehl lpr -rr erfolgen. Welche Druckerprogramme (Backends) möglich sind sowie deren Aufrufparameter entnehmen Sie bitte dem SINIX-Spool Handbuch.

Programmaufruf Parameter

Beim Start der Terminalemulation können folgende Aufrufparameter übergeben werden:

/P*<Name>* Verwendet die angegebene Parameterdatei als Standard-Parameter.

/F Mit diesem Parameter wird zwangsweise in den VGA-Modus geschaltet. Dieser Parameter ist bei dem Monochrom-Bildschirm von Siemens Nixdorf anzugeben.

/V Es wird die Version der Terminalemulation angezeigt, wobei diese nicht gestartet wird.

/M Das Menü der Terminalemulation kann nicht gestartet werden.Folgende Funktionstasten bleiben aktiv bzw. bekommen eine neue Bedeutung:
"Alt+C", "Alt+S", "Alt+X"

"Alt+P" Zeigt die Programmversion.
"Alt+E" Ruft den DOS-Kommandointerpreter auf.

Direktaufruf von Funktionen

Die wichtigsten Funktionen der 97801-Terminalemulation können über folgenden Tastenkombinationen aufgerufen werden:

" Alt + S "	Drucker starten
" Alt + X "	Beenden der Terminalemulation
" Alt + P "	Parametereinstellungen
" Alt + E "	Farbeinstellungen
" Alt + A "	Aktuelle Anwendung
" Alt + C "	Umschalttaste für den ChCode
" Alt + T "	Tastenbelegung ändern